KU LEUVEN

ARENBERG DOCTORAL SCHOOL
Faculty of Engineering Science

Optimal Plasma Edge Configurations for Next-Step Fusion Reactors

Wouter Dekeyser

Dissertation presented in partial
fulfillment of the requirements for the
degree of Doctor in Engineering

February 2014

Optimal Plasma Edge Configurations for Next-Step Fusion Reactors

Wouter DEKEYSER

Supervisory Committee:
Prof. dr. ir. H. Hens, chair
Prof. dr. ir. M. Baelmans, supervisor
Prof. dr. D. Reiter , co-supervisor
(Forschungszentrum Jülich)
Prof. dr. ir. W. D'haeseleer
Prof. dr. S. Poedts
Prof. dr. ir. J. Meyers
Prof. dr. N. R. Gauger
(RWTH Aachen)
Dr. R. A. Pitts
(ITER Organization)

Dissertation presented in partial
fulfillment of the requirements for
the degree of Doctor
in Engineering

February 2014

Celestijnenlaan 300A box 2421, 3001 LEUVEN (Belgium)

D/2014/7515/26
ISBN 978-94-6018-802-2

Preface

Een doctoraat is als een avontuur vol uitdagingen en hindernissen, waar elk overwonnen probleem je de energie geeft om de volgende stap te zetten. De voorbije jaren heb ik genoten van de vele uitdagingen, maar dit werk zou niet mogelijk geweest zijn zonder de vele mensen die mij onderweg geholpen hebben.

In de eerste plaats wil ik mijn promotor, Tine Baelmans, bedanken. Ik heb steeds de vrijheid gekregen om mijn eigen weg te zoeken in dit doctoraat, en kon daarbij altijd rekenen op haar steun, kennis en ervaring. Tijdens de ritten van en naar Jülich en onze vergaderingen die steevast uitliepen zijn talrijke nieuwe ideeën opgeborreld, het ene al meer futuristisch dan het andere. Ik kijk er al naar uit om deze uitdagingen verder aan te pakken.

This work has greatly benefited from the collaboration with the Forschungszentrum Jülich. I am very grateful for the continued support of my co-supervisor, Detlev Reiter, and his invaluable guidance through the complex world of edge plasma physics. His enthusiasm inspired me time and again to face new challenges. I also thank Petra and Vlad for many interesting discussions during the past years. I hope we can continue this fruitful collaboration in the future.

I would like to thank the examination committee for critically reviewing the manuscript, and making valuable suggestions for the finalization of the text.

I greatly enjoyed the past years with my colleagues at TME, the stimulating work environment, the scientific and sometimes less scientific discussions at lunch, and the great TME activities. Also a special thanks to the Mech-runners for helping me explore the forest in Heverlee.

Ook mijn oprechte dank voor Valérie, Frieda en Kathleen die mij steeds geholpen hebben met alle praktische regelingen die in de echte wereld getroffen moeten worden.

Ik ben het Fonds voor Wetenschappelijk Onderzoek - Vlaanderen (FWO) dankbaar voor het toekennen van een doctoraatsbeurs. Tijdens de laatste maanden van mijn onderzoek heb ik kunnen genieten van de rekentijd en ondersteuning verleend door het Vlaams Supercomputer Centrum. Daarnaast wil ik ook prof. M. Van Schoor bedanken om via de Associatie EURATOM-Belgische Staat het kernfusieonderzoek aan de KU Leuven te ondersteunen.

Ook buiten de werkvloer zijn er vele mensen die mij de nodige energie gegeven hebben om mijn doctoraat tot een goed einde te brengen. Mijn ouders, broers en familie wil ik bedanken voor de voortdurende steun en interesse in mijn werk, en mijn vrienden voor de broodnodige ontspanning tussen het werken door.

Tot slot wil ik mijn prachtige vrouw Iona in de bloemetjes zetten, die doorheen dit hele avontuur onvoorwaardelijk aan mijn zijde gestaan heeft, en ons zoontje Daan, die zelfs zonder kernfusie zijn omgeving doet stralen.

Wouter Dekeyser
Heverlee, februari 2014

Abstract

Divertors play a key role in the development of next-step nuclear fusion reactors. Responsible for power and particle exhaust, they have to be designed such that they can safely handle the large power loads. Specifically, their design needs to prevent from exceeding limits imposed by the materials in order to avoid excessive material erosion and melting. At the same time, sufficient particle throughput — in particular Helium pumping capacity — has to be ensured.

In the design process, numerical simulations of the plasma edge are heavily used to assess divertor performance. Typically, plasma edge codes such as B2-EIRENE[1] are used as analysis tools, for example to assist the design of the ITER divertor.[2] Unfortunately, due to the complex nature of the flows and the large number of design variables, extended parametric studies with these edge codes are computationally very demanding, precluding investigating a wide range of divertor geometries and operational points.

Advanced shape optimization algorithms, which have proven their virtue in aerodynamic design, can be particularly useful in partially automating the divertor design process. These methods start from a cost functional which is a measure for the performance of a design, and identify optimal solutions by minimizing this cost functional through adjoint-based optimization algorithms.

This thesis aims at developing efficient, automated design methods for nuclear fusion divertors. State-of-the-art methods from aerodynamics are adapted and extended for use with edge plasma models. Since this work presents the first application of adjoint-based optimization methods to divertors, attention is paid to the general formulation of the design problem as a mathematical shape optimization problem. The methodology is applied to a range of edge plasma models with varying level of detail, focusing on the power exhaust issue.

In the first part of the thesis, focus is on fluid edge plasma models. The

---

[1] Reiter et al., Fusion Sci. Technol. **47** (2005) 172–86.

[2] Kukushkin et al., Fusion Eng. Des. **86** (2011) 2865–73.

continuous adjoint edge plasma equations are derived for a general set of coupled convection-diffusion equations in orthogonal, curvilinear coordinates. The result is directly applicable to the fluid plasma models in codes as B2. Using the velocity method,[3] shape sensitivities are obtained which depend on boundary data only. Therefore, they can be used with any type of grid — in particular with wide grids[4] — and can be evaluated at almost negligible cost. Different optimization algorithms for divertor target design are compared. The one-shot algorithm[5] emerges as the most efficient, providing solutions to the entire design problem at an equivalent computational cost of only 4 to 10 edge plasma simulations. Furthermore, the modularity of this algorithm allows its use with the state solver as it is, facilitating the possible implementation in existing codes. Divertor target shape optimization is then studied both in simplified and in realistic Scrape-Off Layer geometry. In both cases, it is shown that divertor configurations can be obtained with almost perfectly uniform target load. Typically, V-shaped configurations are proposed by the algorithm.

In the second part of the thesis, the methods are extended to coupled fluid-kinetic edge plasma models. In these problems, the kinetic component is treated with a Monte Carlo code, which leads to the additional complexity of statistical noise in the solution algorithm. As a first application, a 1D edge plasma model with a kinetic description of the neutrals is studied. The adjoint kinetic equations are derived, and discussed in view of their implementation in neutral Monte Carlo codes as EIRENE. Convergence of the coupled fluid-kinetic system is studied in detail. It is shown that depending on the parameters and boundary conditions, the model may become very stiff. Using correlated sampling, convergence of the coupled fluid-kinetic system to machine precision can be achieved despite the presence of noise. Furthermore, short cycling leads to a code speedup of a factor of 10 to 100. On the other hand, with uncorrelated sampling and a time averaging procedure, statistical fluctuations can be averaged out effectively. The latter procedure appears very powerful for design optimization, especially in combination with a one-shot approach. The small design updates inherent to this method act as a filter, guiding the design towards its optimal configuration even in the presence of relatively large statistical noise. Finally, a 2D fluid-kinetic edge plasma system is studied in the form of radiation transport coupled to a fluid plasma model. The adjoint radiative transfer equation is derived. By analytically reducing the expressions for the shape sensitivities, reliable sensitivities are obtained which are less prone to Monte Carlo noise. The method is applied to optimize the total divertor target heat load, including the radiation load, showing that a uniform load can be obtained for the test case under consideration.

---

[3] Delfour, *Shapes and Geometries*, SIAM, Philadelphia, 2011.

[4] Baelmans et al., Nucl. Fusion **51** (2011) 083023.

[5] Hazra et al., J. Comput. Phys. **204** (2005) 16–64.

Beknopte samenvatting

In het ontwerp van toekomstige kernfusiecentrales spelen divertoren een essentiële rol. Ze zijn verantwoordelijk voor het veilig en efficiënt afvoeren van het geproduceerde vermogen en de reactieproducten van het fusieproces en moeten in staat zijn om met een extreme warmtebelasting om te gaan. Daarbij mogen ontoelaatbare erosie of smelten van de gebruikte hittebestendige materialen niet optreden.

In het ontwerpproces zijn numerieke simulaties van de plasmarand onmisbaar om het divertorgedrag te voorspellen. Plasmarandcodes als B2-EIRENE[1] worden als analysemiddel gebruikt, bijvoorbeeld bij het ontwerp van ITER.[2] Door het complexe gedrag van het plasma en het groot aantal ontwerpvariabelen is deze procedure echter bijzonder rekenintensief. Praktische beschouwingen laten niet toe om met deze methode een groot aantal ontwerpen grondig te bestuderen.

In de aerodynamica hebben *adjointmethodes* hun intrede gedaan en bewezen dat ze in staat zijn dit type complexe ontwerpproblemen effectief aan te pakken. Met behulp van deze methodes kan het ontwerpproces gedeeltelijk geautomatiseerd worden. Ontwerpdoelstellingen worden opgenomen in een kostenfunctie die op haar beurt geminimaliseerd wordt met efficiënte vormoptimalisatiealgoritmen.

Deze thesis heeft tot doel het ontwerpproces van divertoren te versnellen door het gebruik van geavanceerde adjointmethodes. Aangezien deze methodes voor het eerst toegepast worden op divertorontwerp, wordt uitgebreid ingegaan op de theoretische formulering van het optimalisatieprobleem. In de thesis worden de methodes dan toegepast op een aantal representatieve plasmarandmodellen, met als doel de warmtebelasting van divertoren te optimaliseren.

Het eerste deel van de thesis focust op modellen waarin het plasma als een continuüm beschreven wordt. De adjointvergelijkingen worden afgeleid met behulp van de continue adjointmethode, in een algemene convectie-diffusievorm

---

[1] Reiter et al., Fusion Sci. Technol. **47** (2005) 172–86.

[2] Kukushkin et al., Fusion Eng. Des. **86** (2011) 2865–73.

voor orthogonale, curvilineaire coordinaten. In het bijzonder gelden de uitdrukkingen ook voor de plasmavergelijkingen van B2. Om de vormafgeleiden te bepalen, wordt een snelheidsmethode gebruikt.[3] Met deze methode worden uitdrukkingen bekomen die enkel afhangen van data op de rand van het domein, zodat ze efficiënt geëvalueerd kunnen worden en bruikbaar zijn met elk type grid, en specifiek met zogenaamde *wide grids*.[4] Verschillende optimalisatiemethoden worden met elkaar vergeleken. Een zogenaamd *one-shot*-algoritme[5] blijkt het meest efficiënt. Dit algoritme laat toe het volledige ontwerpprobleem op te lossen in een rekentijd die niet groter is dan 4 tot 10 keer de tijd nodig voor één enkele plasmarandsimulatie. Bovendien is dit algoritme modulair, zodat het gemakkelijk ingebouwd kan worden in een bestaande simulatiecode. Met behulp van de one-shot-methode worden divertoren geoptimaliseerd, zowel in vereenvoudigde als in realistische plasmarandgeometrie. In beide gevallen kan een heel uniforme belasting van de divertor bekomen worden. V-vormige configuraties worden als optimaal naar voor geschoven.

In het tweede deel van de thesis wordt dieper ingegaan op vormoptimalisatie voor gekoppelde continuüm-kinetische plasmarandmodellen. In deze modellen wordt de kinetische component gesimuleerd met een Monte Carlo code, wat op zijn beurt leidt tot statistische ruis in het oplossingsalgoritme. Als eerste toepassing wordt een 1D plasmamodel met een kinetische beschrijving van de neutralen bestudeerd. De adjoint kinetische vergelijking voor de neutralen wordt afgeleid en in detail besproken. Verschillende oplossingsstrategieën voor het gekoppelde systeem worden behandeld, die toelaten de code te versnellen en efficiënt om te gaan met statistische ruis. Door het gebruik van gecorreleerde iteraties kan convergentie tot op machinenauwkeurigheid bekomen worden ondanks de ruis. Het gebruik van gereduceerde continuümmodellen in bepaalde iteraties laat toe de code met een factor 10 tot 100 te versnellen. Een ongecorreleerde koppelingsmethode met tijdsmiddeling is dan weer efficiënt met betrekking tot het uitmiddelen van de ruis. Deze laatste methode wordt ook toegepast om in combinatie met het one-shot-algoritme de geometrie te optimaliseren. Door de kleine ontwerpstappen die intrinsiek zijn aan deze methode wordt de ruis automatisch uitgemiddeld. Ten slotte wordt een 2D model behandeld dat bestaat uit een continuümbeschrijving voor het plasma gekoppeld aan een kinetische beschrijving voor straling. Een adjoint stralingsvergelijking wordt afgeleid. De vormafgeleiden worden op analytisch niveau vereenvoudigd door aan te nemen dat het stralingsprobleem bij benadering isotroop is. Dit leidt tot uitdrukkingen die minder gevoelig zijn voor statistische ruis. De one-shot-methode wordt toegepast om de totale warmtebelasting van de divertor te optimaliseren, inclusief de stralingsbelasting.

---

[3] Delfour, *Shapes and Geometries*, SIAM, Philadelphia, 2011.

[4] Baelmans et al., Nucl. Fusion **51** (2011) 083023.

[5] Hazra et al., J. Comput. Phys. **204** (2005) 16–64.

List of Symbols

Abbreviations

CPU	Central Processing Unit
LCFS	Last Closed Flux Surface
OSD	Optimal Shape Design
PDE	Partial Differential Equation
PFC	Plasma-Facing Component
SOL	Scrape-Off Layer

Constants

e	Elementary charge, $1.60 \cdot 10^{-19}$ C
ϵ_0	Permittivity of free space, $8.85 \cdot 10^{-12}$ F m^{-1}
m_e	Electron mass, $9.11 \cdot 10^{-31}$ kg
m_p	Proton mass, $1.67 \cdot 10^{-27}$ kg

Geometry

Ω	Domain
Σ	Boundary
$\mathbf{e}$	Unit vector
$\sqrt{g}$	Jacobian of metric transformation
h_θ	Metric coefficient, poloidal direction
h_r	Metric coefficient, radial direction
$\boldsymbol{\nu}$	Outward unit normal
$\boldsymbol{\tau}$	Tangential (unit) basis vector

Magnetic field

B	Magnetic field strength (T)
$\mathbf{B}$	Magnetic field vector (T)
b_θ	Magnetic field pitch (–)

Optimization

$\boldsymbol{\mathcal{B}}$	State equations
$\boldsymbol{\mathcal{C}}$	Boundary conditions

$\boldsymbol{\mathcal{B}}^*_{\mathbf{q}}$	Adjoint equations
$\boldsymbol{\mathcal{C}}^*_{\mathbf{q}}$	Adjoint boundary conditions
C	Generalized convective flux vector or coefficient
D	Generalized conduction matrix or coefficient
J	Cost functional, shape functional
$\hat{J}$	Reduced cost functional
L	Lagrangian
$\mathbf{q}$	State variables
$\mathbf{q}^*$	Adjoint variables
$\mathbf{p}^*$	Adjoint boundary variables
S_n	Source terms due to interactions with neutrals
S_p	Plasma source terms
S_z	Source terms due to interactions with impurities
$\mathbf{T}_t$	Design transformation
$\boldsymbol{\mathcal{V}}$	Design velocity

Edge plasma modeling

α_p	Pumping coefficient (–)
c_n	Neutral thermal speed (m s^{-1})
c_s	Isothermal plasma sound speed (m s^{-1})
c_z	Impurity concentration (–)
λ	Decay length (m)
δ_sh	Sheath transmission coefficient (–)
E_i	Ionization energy (J)
E_p	Ionization potential (J)
$E_\mathrm{i,r}$	Ionization energy, radiated part (J)
$E_\mathrm{r,r}$	Recombination energy, radiated part (J)
f_n	Neutral distribution function (s^3 m^{-6})
$\mathbf{\Gamma}$	Particle flux (m^{-2} s^{-1})
I	Radiance or specific intensity (W sr^{-1} m^{-2})
L_z	Radiative loss function (W m^3)
m	Particle mass (kg)
n	Particle density (m^{-3})
p	Pressure (Pa)
Φ_n	Neutral transport flux (s^2 m^{-5})
$\mathbf{Q}$	Energy flux (vector) (W m^{-2})
Q_d	Desired energy flux (W m^{-2})
Q_o	Energy flux to be optimized (W m^{-2})
R	Recycling coefficient (–)
K_cx	Rate coefficient for charge exchange (m^3 s^{-1})
K_i	Rate coefficient for ionization (m^3 s^{-1})
K_r	Rate coefficient for recombination (m^3 s^{-1})
R_E	Energy reflection coefficient (–)

S_E	Internal energy source (W)
S_{E_k}	Kinetic energy source (W)
S_{E_t}	Total energy source (W)
Σ_k	Macroscopic cross-section for interaction k (m^{-1})
$S_{m\mathbf{V}}$	Momentum source ($\mathrm{kg\ m^{-2}\ s^{-2}}$)
S_n	Particle source ($\mathrm{m^{-3}\ s^{-1}}$)
Σ_t	Total macroscopic cross-section (m^{-1})
T	Temperature (J)
u	Velocity component ($\mathrm{m\ s^{-1}}$)
$\mathbf{V}$	Velocity vector ($\mathrm{m\ s^{-1}}$)

Subscripts and superscripts

$'$	Shape derivative
$*$	Adjoint variable
$\cdot$	Material derivative
$\perp$	Diamagnetic direction
θ	Poloidal direction
$\parallel$	Parallel direction
r	Radial direction
ϕ	Toroidal direction
a	Species a
e	Electrons
i	Ions
n	Neutrals
p	Protons

Transport coefficients

D	Diffusion coefficient ($\mathrm{m^2\ s^{-1}}$)
D_p	Pressure diffusion coefficient ($\mathrm{m^2\ s^{-1}\ J^{-1}}$)
κ	Heat conductivity (tensor) ($\mathrm{m^{-1}\ s^{-1}}$)
χ	Radial heat conductivity coefficient (anomalous) ($\mathrm{m^2\ s^{-1}}$)
χ^n	Heat conductivity multiplier for neutrals (–)
η	Dynamic viscosity (tensor) ($\mathrm{kg\ m^{-1}\ s^{-1}}$)
ν	Radial kinematic viscosity (anomalous) ($\mathrm{m^2\ s^{-1}}$)

Contents

Abstract iii

List of Symbols vii

Contents xi

List of Figures xvii

List of Tables xxv

1 Introduction 1

1.1 Tokamak Power and Particle Exhaust 1

1.2 Computational Divertor Design 6

1.3 Optimal Shape Design 8

1.4 Goals and Outline of the Thesis 10

2 Divertor Design through Shape Optimization 13

2.1 Divertor Design as a Shape Optimization Problem 14

2.2 Definition of the Design Space 17

2.3 Basics of Shape Sensitivity Analysis 22

2.3.1 The Material Derivative Approach 22

2.3.2 Shape Derivatives of Domain and Boundary Integrals . 25
2.3.3 Shape Derivatives of Cost Functionals Depending on State Variables 27
2.4 Optimality Conditions 31
2.4.1 First Order Optimality Conditions 31
2.4.2 Design Updates and Gradient Smoothing 33
2.4.3 Continuous versus Discrete Adjoint Approach 34
2.5 Summary 38

3 Elements of the Optimization Problem 39
3.1 Edge Plasma Model 40
3.1.1 Coordinate Systems of the Edge Plasma 40
3.1.2 Plasma Continuity and Parallel Momentum Equations . . 41
3.1.3 Fluid Neutral Model 43
3.1.4 Energy Equation 46
3.1.5 Transport Coefficients 49
3.1.6 Rate Coefficients and Radiative Loss Function 50
3.1.7 Boundary Conditions 51
3.2 General Convection-Diffusion Form of the State Equations . . . 54
3.2.1 State Equations in Poloidal-Radial Coordinate System . 55
3.2.2 State Equations as a Set of Coupled Convection-Diffusion Equations 57
3.3 Derivation of the Adjoint Equations 59
3.3.1 Linearized State Equations 59
3.3.2 Adjoint Equations 61
3.3.3 Adjoint Equations in Poloidal-Radial Coordinate System 66
3.4 Implementation Aspects 66
3.4.1 Implementation of the State Equations 66

3.4.2 Implementation of the Adjoint Equations 74
3.5 Optimization Algorithms 77
3.5.1 Black-Box Methods 77
3.5.2 One-Shot Method 80
3.5.3 Structure of the Optimization Code 82
3.6 Conclusion 83

4 Divertor Shape Optimization in Simplified SOL Geometry 85
4.1 Setup of the Optimization Problem 86
4.1.1 Cost Functional and Shape Parametrization 86
4.1.2 Shape Sensitivity 88
4.2 Numerical Validation 90
4.2.1 Validation of Adjoint Shape Sensitivity Computation . . . 91
4.2.2 Comparison of Optimization Algorithms 93
4.3 Target Shape Optimization in Simplified SOL Geometry 96
4.3.1 Shape Sensitivities 96
4.3.2 Optimized Target Geometries 103
4.4 Conclusion 110

5 Divertor Shape Optimization in Realistic SOL Geometry 111
5.1 Problem Formulation 111
5.2 Grid Generation 112
5.3 Optimization in Realistic Toroidal Geometry 116
5.3.1 Model Problem 116
5.3.2 Density Scan 118
5.3.3 Validation of Shape Sensitivity Computation 124
5.3.4 Divertor Geometry Optimization 125
5.4 Conclusion 133

6 Optimal Shape Design for Coupled Fluid-Kinetic Systems 135

6.1 Kinetic Neutral Gas Modeling 136

6.1.1 Coupled Fluid-Kinetic Edge Plasma Model 136

6.1.2 Solution of the Kinetic Equation with a Monte Carlo Procedure 141

6.2 A 1D Kinetic Neutral Model 145

6.3 Optimization of Target Flux 148

6.4 Derivation of the Adjoint Equations 150

6.4.1 The Linearized Equations 150

6.4.2 The Adjoint Equations 154

6.4.3 Discussion 158

6.5 Design Equation 161

6.6 Conclusion 161

7 Convergence and Optimization of Coupled Fluid-Kinetic Systems 163

7.1 Implementation Aspects 163

7.1.1 Finite Volume Discretization of the Fluid Equations . . 164

7.1.2 Monte Carlo Simulation of the Kinetic Equation 164

7.2 Convergence of Coupled Fluid-Kinetic Models 167

7.2.1 Iterative Relaxation Schemes 167

7.2.2 Uncorrelated Sampling 172

7.2.3 Correlated Sampling 180

7.2.4 Short Cycling 187

7.3 Particle Flux Optimization 190

7.3.1 Validation of Sensitivity Computation 190

7.3.2 Solution of the Optimization Problem 196

7.4 Conclusion 201

8 Optimization of Divertor Radiation Load 203

8.1 Radiation Transport 203

8.2 Optimization of Radiation Transport 207

8.2.1 Linearized Radiation Transport 208

8.2.2 Adjoint Radiation Transport 209

8.2.3 Design Equation 211

8.3 Implementation Aspects 214

8.4 Numerical Validation 217

8.5 Optimization of Total Target Power Load 224

8.6 Conclusion 231

9 Conclusions 233

9.1 General Conclusions 233

9.2 Suggestions for Further Research 237

A Tangential Calculus and Shape Sensitivity Analysis 241

A.1 Tangential Calculus 241

A.2 Shape Sensitivity Analysis 243

B Adjoint Equations in Poloidal-Radial Coordinate System 247

B.1 Derivation Adjoint Field Equations 250

B.2 Derivation Adjoint Boundary Conditions 257

Bibliography 263

Curriculum Vitae 275

List of Figures

1.1 Schematic representation of a tokamak. Figure reproduced from Ref. [93]. 2

1.2 Limiter (left) and divertor (right) configurations of JET. © EFDA-JET 3

1.3 Schematic representation of the SOL. © EFDA-JET 4

1.4 Typical domain of an edge plasma simulation. 8

2.1 Example of divertor geometry parametrization (slab model). . 18

2.2 Mapping the edge geometry to a reference domain. 20

2.3 Middle: geometry 'flowing' along design velocity field $\boldsymbol{\mathcal{V}}$. Right: a specific choice of velocity field $\boldsymbol{\mathcal{V}}(\phi)$ allows to keep the grid field aligned. 22

2.4 Transformation of initial domain Ω to a perturbed domain Ω_t. 23

2.5 Graphical representation of optimal solution. 32

2.6 Continuous versus discrete adjoint approach. Figure based on Refs. [47, 30]. 36

3.1 Definition of different curvilinear coordinate systems. 40

3.2 Temperature dependence of rate coefficients and radiative loss function. 51

3.3 (a) Typical grid for an edge plasma simulation (scaled ITER F57 equilibrium [75], see Chapter 5, Section 5.3). In order to improve visibility, not all grid lines are shown. The thick blue line is the separatrix, the thick black lines represent the cuts. Colors indicate the different boundaries. (b) Topologically rectangular numerical grid. 67

3.4 Five-point discretization stencil for orthogonal grids. The dashed lines represent the poloidal and radial coordinate lines. The solid lines are their discretized counterparts, i.e. the cell faces. Dotted lines are the connector lines between cell centers, which also follow (discretized) poloidal and radial coordinate lines. 70

3.5 Nine-point discretization stencil for non-orthogonal grids. Two cell faces remain aligned with the magnetic field, two are distorted to match the surface. The dashed lines represent the poloidal coordinate lines. Dash-dotted lines are distorted ‘radial’ lines. The solid lines are the cell faces. Dotted lines are the connector lines between cell centers. At each face, two coordinate systems are defined: a normal-tangential system, and a system along and perpendicular to the connector line between cell centers. 72

3.6 Infinitesimal guard cells surround the domain to impose boundary conditions. Figure based on Ref. [5]. 74

3.7 Schematic representation of optimization loop. 82

4.1 (a) Outboard half of ITER divertor and (b) representation of a connected double null divertor in the (θ, r) coordinate system. The dashed line in (b) represents the separatrix. 87

4.2 (a) Shape derivative of reduced cost functional computed with continuous adjoint method (A) and finite differences (FD) on a 130×20 grid, and (b) L1-norm of the difference between both gradients as the grid is refined two resp. four times in each coordinate direction. The dashed line in (b) represents theoretical first order convergence. 92

4.3 Cost functional and L1-norm of the shape derivative for the different optimization methods. 94

4.4 Optimized target profiles and corresponding heat fluxes obtained with the different optimization algorithms. Subscripts ‘0’ refer to the initial, straight target. 95

4.5 (a) Total and desired energy flux to the target, and (b) shape derivative of reduced cost functional computed with continuous adjoint method (A) and finite differences (FD) on a 280×80 grid at initial target geometry. 98

4.6 (a) Initial grid, configuration with straight target. (b) Intermediate configuration, strongly distorted grid. 98

4.7 (a) Total and desired energy flux to the target, and (b) shape derivative of reduced cost functional computed with continuous adjoint method (A) and finite differences (FD) on a 280×80 grid at the intermediate configuration of Fig. 4.6 (b). 99

4.8 (a) Total and desired energy flux to the target, and (b) shape derivative of reduced cost functional computed with continuous adjoint method (A) and finite differences (FD) on a 280×80 grid, using a five-point rather than nine-point discretization scheme. For comparison, also the results obtained with the nine-point stencil are shown in dashed lines (cf. Fig. 4.5). 100

4.9 A comparison of target and midplane profiles computed with five-point and nine-point stencil. 101

4.10 (a) Adjoint gradient before (A) and after (S) smoothing. (b) Target oscillations appearing in one-shot run without gradient smoothing. 102

4.11 Evolution of the residuals, shape sensitivity and cost functional during optimization ($Q_{\mathrm{d,l}}$ case). 104

4.12 Comparison of initial and optimized target profiles and the resulting energy fluxes. The solid lines in (b) are the total energy flux Q_{o}, while the dashed lines represent Q_{t}. The difference between the two is the surface recombination energy, Q_{sr}. . . . 105

4.13 The left figure compares the power deposited along the target for the initial and optimized configurations. On the right, the ratio of initial and desired energy flux densities is shown in dashed lines, and the ratio of optimized and initial surface areas in solid lines. 106

4.14 Comparison of profiles at the targets (figures on the left) and at the midplane (figures on the right) for the initial and optimized configurations. 108

4.15 Comparison of the energy balance of initial (figures on the left) and optimized (figures on the right, $Q_{\mathrm{d,l}}$) target designs. . . . 109

4.16 Comparison of total pressure at upstream (solid lines) and target (dashed lines) locations for initial and optimized cases. Note that for $r < 0$, the upstream and target locations do not correspond to the same flux tube. The high upstream pressure is in the core region, while the low target pressure is in the private flux area. 110

5.1 Poloidal cross-section of ITER (ITER F57 equilibrium [75]), indicating the design parameterization used for the targets. . . 112

5.2 Disk tangent to the flux surface. 114

5.3 Zoom on divertor area of a grid for a (scaled) ITER F57 equilibrium. In order to improve visibility, only every second radial line and every tenth flux surface is shown. The actual grid has 220×240 cells in the poloidal and radial directions, respectively. 115

5.4 Initial configuration: scaled ITER F57 equilibrium [75]. 117

5.5 Target energy loads in initial configuration. The horizontal axis is the distance along the target, with $s = 0$ at the separatrix strike point, $s > 0$ in the SOL and $s < 0$ in the private flux. The solid lines represent the total load (convection, conduction and surface recombination), while the dashed lines contain only convection and conduction. 118

5.6 Outer midplane profiles for different values of the core density n_{c}. The legend gives the value of n_{c} in 10^{19} m^{-3}. 119

5.7 Integrated ion and energy fluxes as a function of core density. IT: flux to inner target, OT: flux to outer target, W: flux to wall, C: flux from core, ion: total (net) volumetric ion source, imp: total energy sink due to impurity radiation, neut: total energy source due to neutral interactions. Fluxes to the private flux boundary are relatively small, and not shown in the figures. 120

5.8 Target fluxes as a function of core density n_{c}. The legend gives the value of n_{c} in 10^{19} m^{-3}. 121

5.9 Maximum density at inner and outer targets as a function of core density. 122

5.10 Profiles of temperature, ion density and neutral density at the targets as a function of core density n_c. The legend gives the value of n_c in 10^{19} m^{-3}. 123

5.11 Shape derivatives computed with continuous adjoint method (A) and finite differences (FD) at both targets. The 'spikes' in the gradients are due to the piecewise linear representation of the targets. 124

5.12 Total target energy loads before and after optimization. Solid lines represent the total energy load, dashed lines the sum of conducted and convected energy. 125

5.13 Comparison of the energy balances of initial (figures on the left) and optimized (figures on the right) target designs. 127

5.14 Target profiles in initial and optimized configurations. 128

5.15 Outer midplane profiles in initial and optimized configurations. 129

5.16 Evolution of the residuals during optimization. 130

5.17 Comparison of angle between target and magnetic field for initial and optimized cases. 131

5.18 Total target energy fluxes as a function of core density n_c for optimized divertor configuration. The legend gives the value of n_c in 10^{19} m^{-3}. 132

5.19 Peak target loads as a function of core density. 132

5.20 Integrated fluxes as a function of core density. Solid lines are for the optimized divertor, dashed lines for the initial configuration (cf. Fig. 5.7). IT: flux to inner target, OT: flux to outer target, W: flux to wall, C: flux from core, ion: total (net) volumetric ion source, imp: total energy sink due to impurity radiation, neut: total energy source due to neutral interactions. Fluxes to the private flux boundary are relatively small, and not shown in the figures. 133

6.1 Sampling from a general cumulative distribution function. Figure based on Ref. [115]. 142

6.2 Sampling from (a) cumulative flight distance distribution and (b) collision distribution. Figure based on Ref. [115]. 143

7.1 Representation of the 1D grid. Collocated cells are indicated with solid lines, staggered momentum cells with dashed lines. . 164

7.2 Number of iterations required for convergence as a function of recycling coefficient. In figure (a), the core density is increased as R increases (see Table 7.1). In figure (b), core density is the same for all values of R. 170

7.3 Profiles of state variables for different values of R. 171

7.4 (a) Adjoint density and neutral transport flux , and (b) adjoint parallel velocity. 172

7.5 Influence of stochastic noise on the residuals of the plasma equations. The residuals fail to decrease due to the use of new random numbers in every iteration. 173

7.6 Time trace of ion and neutral density in a cell at $\theta = 0.09$ m. . 174

7.7 (a) Residuals $R_{\mathrm{c}}^{*}(\phi^{*})$ and $R_{\mathrm{m}}^{*}(\phi^{*})$ of adjoint continuity and momentum equations in case of uncorrelated sampling. (b) $R_{\mathrm{c}}^{*}(\overline{\phi}^{*})$ and $R_{\mathrm{m}}^{*}(\overline{\phi}^{*})$ as a function of iteration number. 175

7.8 (a) Adjoint ion density at final iteration (final), averaged over all iterations (avg) and computed with the finite volume kinetic model (FV). (b) Adjoint neutral transport flux at final iteration (final), averaged over all iterations (avg) and computed with the finite volume kinetic model (FV). Solid lines in (b) represent $\Phi_{\mathrm{n},+}^{*}$, dashed lines represent $\Phi_{\mathrm{n},-}^{*}$. 176

7.9 (a) Residuals $R_{\mathrm{c}}(\phi)$ and $R_{\mathrm{m}}(\phi)$ of continuity and momentum equations in case of uncorrelated sampling. (b) $R_{\mathrm{c}}(\overline{\phi})$ and $R_{\mathrm{m}}(\overline{\phi})$ as a function of iteration number. 177

7.10 Level of the residuals $R_{\mathrm{c}}(\phi)$ and $R_{\mathrm{m}}(\phi)$ (solid lines) and $R_{\mathrm{c}}(\overline{\phi})$ and $R_{\mathrm{m}}(\overline{\phi})$ (dashed lines) as a function of number of Monte Carlo histories. The black lines represent the n_{t}^{-1} and $n_{\mathrm{t}}^{-1/2}$ slopes. . 178

7.11 A comparison of time-averaged profiles obtained for varying n_{t} with finite volume neutral solution. The error bars represent three times the standard deviation of time fluctuations. Figure (d) shows the reduction of the error (solid lines) compared to the finite volume solution as well as the L2-norm of the standard deviation (solid lines with markers). 179

7.12 In case of correlated sampling, the residuals of the fluid equations go down to machine precision. 181

7.13 A comparison of profiles computed with finite volume and Monte Carlo solvers for the neutrals. On the left, profiles are shown for single runs with 10^2, 10^3 and 10^4 trajectories. The figures on the right compare the corresponding average profiles over 500 samples. The error bars represent three times the local sample standard deviation. 183

7.14 The reduction of (a) the L2-norm of error compared to finite volume solution (solid lines) and sample standard deviation (lines with markers) and (b) the L1-norm of the truncation error. The dashed lines represent the $n_\mathrm{t}^{-1/2}$ and n_t^{-1} slope, respectively. . 185

7.15 Residuals of the plasma equations when short cycling is used with correlated sampling. Intermediate 'peaks' in the residuals indicate iterations where a Monte Carlo run is performed. . . 189

7.16 (a) Cost functional as a function of domain length and (b) shape sensitivities computed with adjoint (A) and finite difference (FD) methods. 191

7.17 Individual components of the shape sensitivity. 191

7.18 Evolution of the time-averaged shape sensitivity for different numbers of trajectories per iteration. 194

7.19 Evolution of the residuals of the time-averaged solution during sensitivity computation with uncorrelated sampling. 195

7.20 Some examples of the evolution of the design variable during one-shot optimization with correlated sampling ($n_\mathrm{t} = 10^2$). . . 199

7.21 (a) Evolution of design variable during one-shot optimization. (b) Standard deviation of the oscillations on L_θ as a function of number of histories n_t. (c) Value of the the cost functional and (d) evolution of the time-averaged shape sensitivity during optimization. 200

7.22 History of optimization with $n_\mathrm{t} = 10^3$. After 10.000 iterations, time step and design relaxation factor are reduced by an order of magnitude. 201

8.1 Reflection models. 206

8.2 Computation of intersection between photon trajectory and a vessel segment. In the poloidal plane, the photon trajectory does not necessarily appear as straight. 216

8.3 (a) Geometry of the test radiation test problem. (b) Simulated and desired fluxes for which the shape sensitivities are assessed. 218

8.4 Comparison of continuous adjoint and finite difference shape sensitivities, (a) using 10^7 histories and (b) using 10^8 histories. Figure (c) shows the finite difference gradient of figure (b), but filtered 10, 100, and 1000 times with a top hat filter. In figure (d), the individual components of the adjoint shape sensitivity are shown (10^7 histories). 221

8.5 One-shot optimization with gradient smoothing. 222

8.6 One-shot optimization with gradient filtering and smoothing. . 224

8.7 Geometry used for the simulation of the plasma and radiation. Details of the dome are neglected. The plasma grid is colored in red. The vessel segments are colored according to the material used. Black: Carbon, blue: Tungsten, green: Beryllium. 225

8.8 Total energy fluxes to targets for scaled ITER F57 divertor (label: ‘initial configuration’), and design optimized in Chapter 5 (label: ‘optimized configuration’). The dashed lines represent convected and conducted energy, the dash-dotted lines include the contribution of surface recombination and the solid lines additionally include the radiative load. 226

8.9 Radiation sources in initial configuration (left figures) and in configuration optimized in Chapter 5 (right figures). 227

8.10 History of residuals during one-shot optimization. 228

8.11 Initial and optimized total target energy fluxes. Dashed lines represent the target energy without contribution from radiation. 229

8.12 Optimized divertor geometries, (a) without taking into account radiation (see Chapter 5) and (b) with radiation included in the cost functional. 230

A.1 Local parametrization of the domain. Figure based on Ref. [29]. 242

List of Tables

1.1 Heating power (P_{heat}), major radius (R), and characteristic numbers of divertor loading for some current, planned, and next-step devices. Table reproduced from Ref. [74]. 7

4.1 Comparison of CPU times required for shape sensitivity computation. 93

4.2 Parameters of the JET-sized model problem. 97

4.3 Comparison of ion particle and energy balances at initial and optimized configurations. 107

5.1 Model parameters and boundary conditions. 117

5.2 Comparison of particle and energy balances at initial and optimized configurations 126

7.1 Model parameters. 169

7.2 Time step Δt (s) and number of iterations N required for convergence. 170

7.3 Global particle balance as a function of number of Monte Carlo particles. 180

7.4 Accuracy of shape sensitivity computation with Monte Carlo neutrals and correlated sampling. The average sensitivity and standard deviation are given for different numbers of histories. Since the second component of the gradient is zero due to the choice of L_0, it is not shown in the table. 193

7.5 Accuracy of shape sensitivity computation with Monte Carlo neutrals and uncorrelated sampling. The values of the shape sensitivities are averages over 10000 iterations. The standard deviations are those of the time fluctuations. Since the second component of the gradient is zero due to the choice of L_0, it is not shown in the table. 194

7.6 Accuracy of Monte Carlo shape sensitivity computation using finite differences, with (CS) and without (FD) correlated sampling. For reference, also the uncertainty on the cost functional is indicated. 196

7.7 Time step Δt (s), gradient relaxation factor r (-) and number of optimization iterations n for fastest convergence of the optimization problem. 197

8.1 Reflection properties of some commen fusion reactor materials. Table based on Ref. [4]. 207

8.2 Comparison of balances at initial and optimized configurations. The middle column refers to the configuration optimized in Chapter 5 (see Table 5.2). 230

Chapter 1

Introduction

With a clean, cheap, widely available and virtually inexhaustible fuel, nuclear fusion has the potential to provide a sustainable answer to our energy needs. However, several issues have to be resolved in order to enable its commercial realization.

In magnetic confinement fusion, the tokamak principle is presently the most advanced. With ITER, a 500 MW tokamak currently under construction in Cadarache, France, the leap from experimental-scale to reactor-scale machines is made, and the establishment of computational engineering and design concepts for DEMO fusion power plant components is an urgent topic. This thesis focuses on the important aspect of reliable and efficient power exhaust from the reactor, as of today an unresolved issue.

1.1 Tokamak Power and Particle Exhaust

In a tokamak, the fusion plasma is confined by strong magnetic fields, Fig. 1.1. A toroidal magnetic field is generated by toroidal field coils. By inducing a toroidal current in the plasma with a central transformer, a poloidal magnetic field is generated. Additional poloidal field coils allow for plasma shaping and control. This leads to helical magnetic field lines which form a set of closed, nested magnetic flux surfaces. Due to the Lorentz force, the ions and electrons in the plasma are forced to gyrate around the field lines, thereby effectively confining the plasma to the flux surfaces. On the other hand, particle motion along the field is virtually unrestricted.

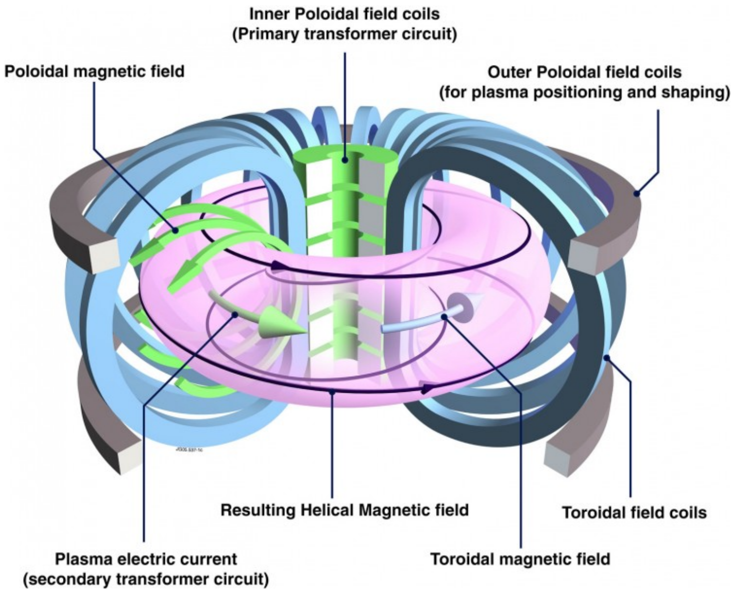


Figure 1.1: Schematic representation of a tokamak. Figure reproduced from Ref. [93].

Due to collisional and turbulent processes, slow cross-field transport will eventually transport ions (Deuterium and Tritium) and electrons from the core towards the edge. In order to protect the vessel from the plasma, limiters or divertors can be used, see Fig. 1.2. In a limiter tokamak, the size of the plasma is restricted by the introduction of a solid object into the edge plasma. Such a *limiter* defines the Last Closed Flux Surface (LCFS) separating the core plasma from the Scrape-Off Layer (SOL). Once the particles diffuse across the LCFS, the fast transport along the magnetic field rapidly takes the particles to the limiter, before there is much time to propagate further radially outwards. The width of the SOL can be estimated by considering the time scales for parallel and radial transport [116]. As the plasma travels over a distance L from an upstream position to the target (the connection length), it accelerates from speed zero upstream to sound speed c_s due to the Bohm criterion. The typical transit time is thus

$$\tau_\mathrm{SOL} \sim \frac{L}{c_\mathrm{s}}.$$

In the cross-field direction, plasma transport is governed by turbulent processes, typically modeled by a diffusion coefficient $D_\perp$. The characteristic distance traveled in the radial direction is then $\lambda_{\mathrm{SOL}} \sim \sqrt{D_\perp \tau_{\mathrm{SOL}}}$, or

$$\lambda_{\mathrm{SOL}} \sim \sqrt{\frac{D_\perp L}{c_{\mathrm{s}}}}.$$

λ_{SOL} is a good estimate of the width of the SOL. For example, using the representative values $D_\perp \approx 1$ m^2 s^{-1}, $L \approx 50$ m and $c_{\mathrm{s}} \approx 5 \cdot 10^4$ m s^{-1}, the width of the SOL is only a few centimeter.

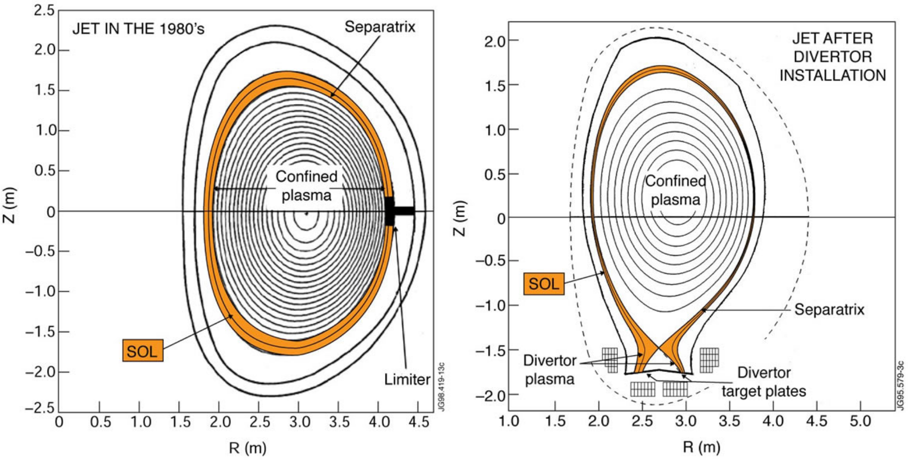


Figure 1.2: Limiter (left) and divertor (right) configurations of JET. © EFDA-JET

In a divertor tokamak, additional external coils are used to create a magnetic X-point (see Figs. 1.2 and 1.3). A separatrix defines the boundary between the core plasma or confinement region and the SOL. The plasma is now terminated by the introduction of targets plates in the diverted plasma. Again, the competition between fast parallel and slow radial processes effectively limits the radial width of the plasma.

When ions and electrons strike a solid surface (limiter, target plates, or the vessel wall), they recombine and form neutral atoms and molecules. These neutrals are no longer confined by the magnetic field, and are free to move back into the plasma or towards the pumps. The neutrals that are not pumped out eventually get ionized again by collisions with magnetized ions or electrons, and become part of the plasma in a process called *recycling*.

Both limiters and divertors limit the size of the plasma. However, the interaction of the plasma with the vessel is now concentrated on a rather small part of the vessel. These components are thus prone to severe material erosion, and must be

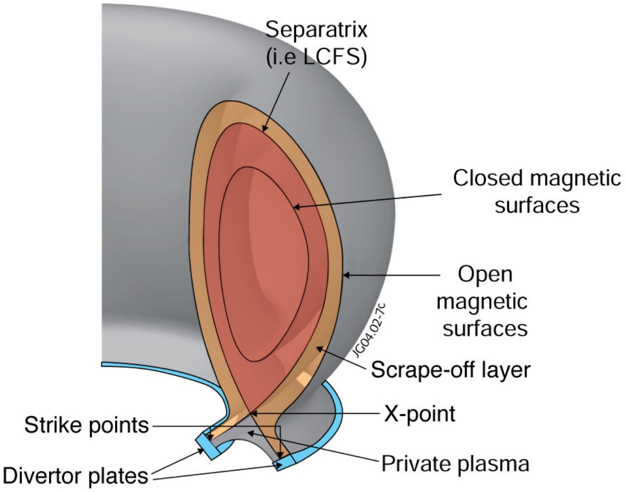


Figure 1.3: Schematic representation of the SOL. © EFDA-JET

designed specifically to withstand the extreme power densities, accompanying temperatures and bombardment by ions and recycling neutrals. Furthermore, nuclear safety requirements pose limits on the the amount of radioactive Tritium that can be retained in these materials. Both concepts have advantages and disadvantages. For a detailed comparison, see for example Refs. [116, 36]. Here, the aspects most relevant for the present work will be highlighted.

The earliest tokamak machines were limiter devices, while divertors appeared in the eighties. Today, almost all tokamaks have a divertor installed. The original motivation to use a divertor tokamak is the reduction of the impurity sources and impurity contamination of the main plasma. By moving the main source of impurities (typically in the area of most intense plasma-surface interaction) to the divertor targets, the impurities have to travel a longer way through the SOL before they reach the core plasma. In principle, the divertor targets can be moved as far as needed. With a limiter configuration, this freedom is not available. In fact, there the area of strongest plasma-wall interaction is around the leading edge of the limiter, i.e. almost in direct contact with the confined plasma.

Another important advantage of the divertor is the Helium pumping capability. Helium appears as the waste product of the fusion reaction, and has to be removed to avoid fuel dilution. By compressing the plasma and neutrals in the divertor legs, locally rather high neutral Helium pressures can be achieved, and comparatively small and efficient pumps can be installed in the divertor area. Furthermore, efficient pumping systems not only allow for the removal of Helium, but also for the control of plasma density through the active pumping of Hydrogen.

Due to the rather long connection length L between upstream positions and the target, a divertor allows to combine high upstream temperatures compatible with good core confinement and low target temperatures, with resulting low impurity sources. Furthermore, a lot of power can be removed by volumetric processes as hydrogenic or impurity radiation and charge exchange with neutrals. To this end, low Z (low charge-state) elements as N or Ne may be seeded. Due to these processes, the power is deposited more uniformly over a larger area of the divertor.

By installing and using divertors, two very beneficial operating regimes have been discovered: the H-mode (high confinement mode) and plasma detachment. The H-mode — a regime of improved energy confinement — is found at sufficiently high power. While the transition to H-mode is not yet understood theoretically, it is believed that the reduction of turbulent radial transport in the edge to neoclassical levels and the formation of an edge transport barrier lie at its origin [36]. This results in high core pressure and temperature and good energy confinement. By increasing the density or the volumetric power losses, a detached regime can be achieved. In this regime, the combination of plasma momentum losses due to frictional drag with neutrals and volumetric power losses lead to very low values of target temperature and plasma density, thus effectively detaching the plasma from the wall.

The divertor has a few inherent disadvantages compared to limiter tokamaks, starting with the complexity and cost. Indeed, additional coils are needed to create the magnetic X-point configuration. Furthermore, the divertor takes up a significant fraction of the magnetic volume, thereby reducing the volume available for the fusion process itself. Another important issue is the plasma-wetted area. By concentrating the plasma-surface interactions on a small part of the vessel, the area available for SOL power depositions is much smaller than the total area of the vessel. As a result, the power exhaust issue is very critical for divertors, especially for reactor-scale devices. In a limiter, on the other hand, with a bumper limiter and careful control of the magnetic configuration, theoretically almost the entire inner wall is available for power deposition. Despite these considerations, the achievement of the H-mode regime with divertor tokamaks and the improved pumping efficiency largely outweigh the disadvantages and explain their current popularity.

1.2 Computational Divertor Design

Designing divertors that can safely handle the power and particle exhaust of next-step fusion reactors is a challenging task. A large fraction of the fusion power has to be extracted through the divertor, while power loads to plasma-facing components should not exceed limits imposed by the materials in order to prevent excessive erosion and melting. Experimental limits of ~ 10 MW m^{-2} for steady-state heat loads, and ~ 0.5 MJ m^{-2} in 250 μs for transient loads (e.g. Edge Localized Modes or ELMs) are found for common target materials [36]. At the same time sufficient pumping capability for the Helium ash has to be ensured. In order to give a proper perspective, some representative values of the heating power for current, planned and next-step devices are summarized in Table 1.1, reproduced from Ref. [74]. The first four devices in the table are present-day experimental devices. ITER is currently under construction in Cadarache, France. The other machines are conceptual DEMO reactor studies. For details, see Ref. [74] and references therein. The last three columns contain some metrics describing the intensity of the divertor power load, normalized to their value for ITER. The metrics differ in the assumption on how the SOL width (and thus plasma-wetted area) scales with major radius. If the SOL width is assumed to be constant, the metric P_{heat}/R is obtained. The other metrics assume linear and quadratic increase of the SOL width with major radius, which may be the appropriate scaling for strongly radiating divertors [74]. In any case, the average loading of reactor-scale divertors will be significantly higher than for ITER, which in itself is already considered as pushing the limits of standard divertor concepts. In order to keep heat loads to the high-heat-flux components in fusion reactors below acceptable levels, it is generally accepted that a significant fraction (over 50%) of the power entering the SOL will have to be radiated by impurities, while the divertor will have to operate in a (partially) detached regime. Furthermore, advanced divertor concepts such as the X-divertor [74], super-X divertor [124] and snowflake divertor [106, 107] may have to be considered.

Presently, divertor design is heavily assisted by numerical simulation tools. Typically, ions and electrons are modeled with Navier-Stokes-like multi-fluid equations for particle, momentum and energy conservation, while neutrals require a kinetic treatment using Monte Carlo codes. Collision processes between the different particles (e.g. ionization, recombination, charge exchange) then lead to source or sink terms in the various fluid equations. A typical simulation domain for a divertor is shown in Fig. 1.4. Toroidal symmetry is assumed, so the edge plasma can be described in the poloidal plane. In the poloidal direction, the domain extends between the two divertor targets (AB) to (CD). The interaction with the core plasma is modeled through boundary

Table 1.1: Heating power (P_{heat}), major radius (R), and characteristic numbers of divertor loading for some current, planned, and next-step devices. Table reproduced from Ref. [74].

Device	P_{heat} (MW)	R (m)	P_{heat}/R ITER = 1	P_{heat}/R^2 ITER = 1	P_{heat}/R^3 ITER = 1
C-Mod	3	0.6	0.26	2.7	—
DIII-D	10	1.6	0.31	0.68	—
JET	17	3	0.31	0.60	—
JT-60U	17	3.4	0.26	0.55	—
ITER	120	6.2	1	1	1
EU-A	1246	9.6	6.8	4.3	2.8
EU-B	990	8.6	6.1	4.3	3.2
EU-C	792	7.5	5.6	4.5	3.8
EU-D	571	6.1	4.9	4.9	5.0
ARIES-AT	387	5.2	3.9	4.6	5.6
ARIES-RS	515	5.5	4.9	5.4	6.2
Slim-CS	645	5.5	6.2	6.8	7.8
CREST	691	5.4	6.7	7.6	8.8

conditions at the edge-core interface. Indeed, due to the low density of the edge plasma compared to the core, its effect on the magnetic equilibrium is neglected and the full MHD equations are not solved by the edge codes. This approximation is not valid in the core. At the first wall (BC) and private flux (AD) boundaries, the simulated domain usually extends to the last flux surface tangent to the first wall or dome, respectively.

Examples of the resulting coupled fluid-kinetic codes are B2-EIRENE [96], EDGE2D-NIMBUS [113], EDGE2D-EIRENE [128], and UEDGE-DEGAS [101, 117]. The different edge codes may differ in particular aspects of the implementation. Specifically, different versions of B2-EIRENE (SOLPS) use closure schemes for the fluid equations based on either Braginskii [16] (SOLSP4.x) or Balescu [9] (SOLPS5.x). Drifts and currents have been added to B2 in the nineties [5, 8, 7], with the most complete model at present in SOLPS5.2 [105, 103]. Extension of the simulated domain all the way up to the first wall with so-called *wide grids* is treated in Refs. [6, 24].

By detailed physical modeling, these hybrid fluid-kinetic codes allow to extrapolate based on the current knowledge of edge plasma physics to fusion-reactor-relevant parameter windows which are, as of today, experimentally inaccessible. In particular, the B2-EIRENE code is heavily used to assist the design of the ITER divertor [76, 77, 75]. Also for DEMO reactor studies and the

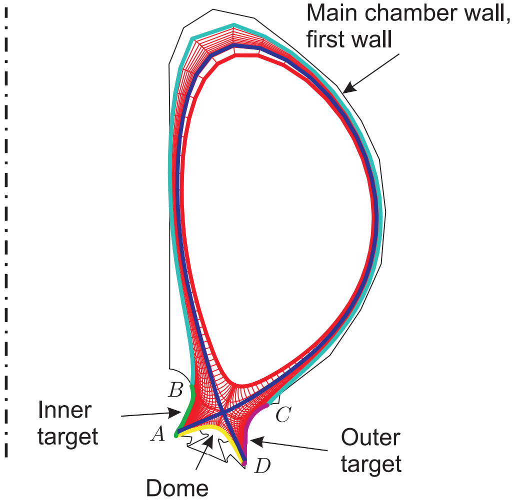


Figure 1.4: Typical domain of an edge plasma simulation.

assessment of novel divertor concepts, numerical tools are indispensable [88, 104]. Typically, the edge codes are used in a design-by-analysis approach, where divertor configurations and magnetic fields are manually adjusted to meet design requirements. However, due to the different design requirements and constraints combined with the large number of control variables and the complex nature of the flows, this method is extremely demanding, both computationally and in terms of manpower. Typical run times for ITER are in the order of a few months. When scaling up to DEMO-relevant parameters, simulations may take up to a year. The long simulation times preclude investigating many different geometries and operating points during the design process.

1.3 Optimal Shape Design

Similar design challenges are often faced in aerodynamics. There, the aim may be to design airfoils for a given pressure profile, for drag reduction or optimal lift-to-drag ratio, to increase the pressure gain in compressor stages, or to shape ducts for minimum pressure or viscous losses. In these applications, the flows are constrained by the Navier-Stokes equations. These problems have been treated very effectively by using optimization approaches where the shape is considered as the design or control variable. The design objective is formulated as a cost functional, which is to be minimized by changing the shape of the air foil or duct. The Navier-Stokes equations are introduced as a constraint on the flow field. Using optimization algorithms, the best design for a specific design objective is automatically obtained, while ensuring that the model equations are satisfied at the optimum design point.

In the literature, a long history of optimal control for systems governed by partial differential equations (PDEs) can be found. The pioneering work by J. L. Lions [79] is mentioned. Optimal shape design (OSD) is a special branch of control problems where the control variable is the shape of the domain. Over the past decades, OSD has been applied to a variety of problems. For a detailed overview of applications and developments in fluid mechanics, see Ref. [83], and specifically for aerodynamics, see Ref. [43]. Optimal shape design for Stokes flow [91] and for elliptic systems [92] was treated very early by O. Pironneau. The first to apply these methods to aerodynamic shape design in transonic flows was A. Jameson, first for the control of potential flow [62], and later also for the Euler equations [63] and the Navier-Stokes equations [65]. Applications from 2D airfoils [62, 1] to 3D wing and wing-body configurations [98] have been studied, as well as complex geometries and complete aircraft [99, 17]. Other aerodynamics application include the reduction of viscous losses for internal flows and blade design for turbomachinery [89, 43].

An essential feature of shape optimization problems constrained by partial differential equations is the huge cost associated with sensitivity analysis if straightforward finite difference approaches are used. Indeed, after discretization or parameterization of the shape, a large number of design variables results. Computing the gradient of the cost functional at a certain design point using finite differences requires perturbing each design variable in turn, and solving the governing model equations to determine its influence on the cost functional. This cost quickly becomes prohibitive. An adjoint or dual set of PDEs can be derived, however, which contains the necessary information on how to change all design variables in order to obtain a reduction in the cost functional. The adjoint problem typically has similar structure as the forward analysis problem, but has reversed characteristics. It can be solved at a comparable computational cost. Thus, independent of the number of design variables, the gradient of the cost functional with respect to all design variables can be computed at an equivalent cost of approximately two flow simulations, i.e. one solution of the forward system and one of the adjoint system. This gradient can then be used in traditional gradient descent algorithms.

Alternatively, the forward model equations, the adjoint equations, and the equation stating that the gradient of the cost functional must be zero at the optimum (the design equation) may be regarded as one large system of coupled (partial differential) equations, and solved simultaneously. Such methods are often called one-shot methods, a term which appears to have been introduced by Ta'asan [120, 78]. Several authors have successfully used different variants of these methods in the field of aerodynamic design [55, 57], often in the context of a multigrid approach [2, 53, 12]. Using these methods, the complete shape optimization problem can be solved at an equivalent cost of roughly 5 to 10 forward flow simulations.

While excellent overall performance can be achieved with a one-shot method, the success of this approach depends on the performance of all individual elements in the design code. Indeed, since state, adjoint and design equations are iteratively solved as one coupled system, each of the equations has to be updated several thousand times. Due to the specific nature of the design variable in optimal shape design, the mapping from shape to flow problem involves an (in practice often very complex) grid generation step. If this part of the code cannot be handled by an adjoint approach, grid generation may become a bottleneck for the one-shot method. On the theoretical side, significant progress is made on the computation of derivatives of flow solutions with respect to the shape of the domain in the framework of shape calculus and shape sensitivity analysis [114, 52, 29]. In this framework, it can be shown rigorously that shape sensitivities depend only on boundary data, and can thus be evaluated at a very low cost. Based on this result, truly large-scale applications with tens of thousands of design variables have been treated with a one-shot approach in Ref. [108].

Apart from having an automated and highly efficient design procedure, working in an optimization framework offers another important advantage. The optimization environment naturally allows for the incorporation of constraints. These constraints may be direct constraints on the design variables, or may involve additional constraints on the flow variables as in Refs. [56, 54].

1.4 Goals and Outline of the Thesis

The aim of this thesis is to develop efficient, automated design methods for nuclear fusion divertors. These methods must be able to handle the specific challenges of the divertor application. First of all, given the very large computational cost of edge plasma simulations in reactor-relevant regimes, a prime criterion for the selection of the optimization algorithms will be the cost in terms of number of flow simulations. Furthermore, the methods must cope effectively with the coupled fluid-kinetic nature of edge plasma models. In particular, they have to be robust with respect to statistical noise due to a Monte Carlo code for the kinetic components. Apart from efficiency and robustness, the method must be able to take into account the typical constraints appearing in divertor design applications. As a final consideration, in order to ensure penetration of the methodology into the plasma edge community, portability to existing edge codes is an important criterion.

Secondly, this thesis aims to apply the developed methods to a key performance issue for future divertors: the power exhaust problem. The aim is to reduce the peak heat loads to divertor targets, and spread out the power as much as possible along the high-heat-flux plasma-facing components (PFCs) by carefully controlling the target shape. At the same time, (Helium) pumping capabilities have to be maintained.

Motivated by the success of adjoint-based optimization methods in aerodynamics, these methods will be pursued in this thesis. In nuclear fusion applications, adjoint methods are used for example in magnetic field reconstruction [15]. They have also been applied to the characterization of thermal properties of surface layers in [39]. In edge plasma applications, the automated evaluation of transport coefficients with an optimization approach (but not with adjoint sensitivity computation) has been used in [22]. To the knowledge of the author, this thesis presents for the first time the use of an adjoint edge plasma model for efficient (shape) sensitivity computation and an automated approach to divertor design. Therefore, attention will also be paid to the underlying theoretical concepts and terminology.

Chapter 2 starts with a theoretical formulation of the design problem as a mathematical shape optimization problem. Suitable shape parametrizations and the computation of sensitivities of the cost functional with respect to the shape of the domain are investigated. Finally, optimality conditions which must be satisfied by the optimal solution are formulated.

In Chapters 3 to 8, the general theory presented in Chapter 2 is applied to somewhat simplified edge plasma models with varying degree of sophistication, which, however, capture the most essential ingredients and nonlinear challenges of current divertor design codes. This part of the thesis is divided in two main parts. In Chapters 3 to 5, divertor shape optimization is studied for problems governed by fluid edge plasma equations. In particular, neutrals are not treated kinetically, but a fluid pressure diffusion equation is assumed instead. This allows to focus on the details of the optimization algorithms without the additional complexity of statistical noise due to Monte Carlo solvers. The model is presented in Chapter 3. The chapter then continues with a derivation of the adjoint edge plasma equations and a discussion of the optimization algorithms. Finally, an overview of the structure of divertor design code is given. In Chapter 4, the divertor shape optimization algorithms are applied in simplified SOL geometry. A ‘slab’ representation of the SOL is assumed. This allows to construct relatively easy test problems. A thorough numerical validation is performed, and different optimization algorithms are compared. The chapter concludes with an in-depth study of the potential and performance of optimal target design. Chapter 5 considers divertor optimization in realistic edge plasma geometry. It is shown that the algorithm can handle geometrically complex, ITER-like divertor configurations. Furthermore, the optimized divertor configuration is obtained at an equivalent computational cost of only 10 simulations, proving the potential of the automated design method.

In the second part of the thesis, the scope is broadened to coupled fluid-kinetic edge plasma models. First, Chapter 6 considers problems where the neutral plasma component is described by a kinetic equation. The derivation of the adjoint kinetic equations and shape sensitivities is presented. In order to simplify the problem in a first exploratory step, a 1D coupled fluid-kinetic model is studied. However, the resulting equations are discussed in view of their potential implementation in a complete 2D edge code. In Chapter 7, the theory developed in Chapter 6 is tested in practice. First, a detailed study of the convergence behavior of fluid-kinetic edge plasma models is performed. The performance of different code coupling schemes is assessed. Furthermore, since the kinetic neutral equation is solved with a Monte Carlo procedure, statistical noise is introduced in the residuals of the fluid equations, posing a challenge on the convergence of the latter equations. Therefore, also several schemes to handle this noise are analyzed in terms of speed and accuracy. Then, all ingredients are at hand to study shape optimization of fluid-kinetic systems at the end of Chapter 7. Finally, in Chapter 8 a first 2D fluid-kinetic system is studied in the form of radiation transport coupled to the plasma. It is shown that under certain assumptions, the radiation problem can be treated in a post-processing step. The adjoint radiation equations and shape sensitivities are derived. After a numerical validation, the realistic divertor shape optimization problem from Chapter 5 is revisited, this time including radiation transport.

As a recurring theme, all model problems and test cases in the different chapters consider the divertor power exhaust problem. In order to avoid extreme heat loads as much as possible and to optimally profit from the high-heat-flux components, the optimization goal is to achieve as uniform a target power load as possible. In the first part of the thesis, only the energy transported through the SOL is considered. In Chapter 8, also radiative energy is added to the picture.

In the final chapter, general conclusions are formulated and suggestions for further research are given.

Chapter 2

Divertor Design through Shape Optimization

This chapter deals in general terms with a broad class of divertor design problems that can be tackled by shape optimization methods. Since adjoint methods for shape optimization are applied for the first time to edge plasma applications, attention is paid to the formulation of the mathematical optimization problem and the introduction of the necessary tools and concepts. In Section 2.1, the general formulation of the divertor design process as a shape optimization problem is elaborated. A large body of literature on shape design is available, mainly from the field of aerodynamic design. Depending on the needs of the application, different methods for shape parametrization and shape sensitivity analysis have been developed. A brief overview of the methods most relevant to divertor design is given in Section 2.2. Next, Section 2.3 goes into the details of the nonparametric approach, which appears as the most general, most promising candidate for edge plasma applications. The chapter ends with the presentation of first order optimality conditions which must be satisfied by the optimal design.

The material presented in this chapter serves as a starting point for the PhD research. More details on the methods used can be found in the literature and in several text books, for example Refs. [59, 121, 13] for optimization with PDE constraints and Refs. [114, 52, 19, 29] for shape sensitivity analysis.

2.1 Divertor Design as a Shape Optimization Problem

In order to formulate the divertor design problem as a shape optimization problem, design objectives are incorporated in a cost functional or objective functional $J(\Omega, \mathbf{q})$. This cost functional measures the performance of a design with respect to the selected objectives. The lower the value of the cost functional, the better the design.

The cost functional depends on two types of variables. First of all, and typical for shape optimization problems, it depends on the (shape of the) domain Ω on which the problem is defined. This is the independent variable, *control* variable or *design* variable in the optimization problem. In particular, cost functionals are considered which are integrals over the domain Ω or (a part of) its boundary $\Sigma \equiv \partial\Omega$, for example

$$J_1(\Omega, \mathbf{q}) \quad = \quad \int_\Omega f\left(\mathbf{q}\right) \,\mathrm{d}\boldsymbol{\omega}, \tag{2.1}$$

$$J_2(\Omega, \mathbf{q}) \quad = \quad \int_\Sigma g\left(\mathbf{q}, \boldsymbol{\nu}\right) \,\mathrm{d}\sigma, \tag{2.2}$$

with f and g sufficiently smooth functions. $\boldsymbol{\nu}$ is the outward pointing unit normal field. Secondly, the cost functional also depends on the vector of plasma state variables $\mathbf{q}$. Each component of $\mathbf{q}$ is typically a function defined on Ω, e.g. the plasma density, temperature and so on. These state variables in turn depend on the shape of the domain through the plasma *state equations* $\boldsymbol{\mathcal{B}}(\Omega, \mathbf{q}) = 0$, usually a system of partial differential equations (PDEs) with corresponding boundary conditions for each of the state variables. They could be the full set of Braginskii equations [16] augmented with a (kinetic) model for the neutrals, for example, or somewhat simplified models as used in this thesis. In order to emphasize the complex dependence of the state equations on the domain, $\boldsymbol{\mathcal{B}}(\Omega, \mathbf{q})$ is written explicitly as field equations[1] $\boldsymbol{\mathcal{B}}(\mathbf{q})$ with boundary conditions $\boldsymbol{\mathcal{C}}(\mathbf{q}, \boldsymbol{\nu})$:

$$0 = \boldsymbol{\mathcal{B}}(\Omega, \mathbf{q}) = \left\{ \begin{array}{ccc} \boldsymbol{\mathcal{B}}(\mathbf{q}) & \text{in} & \Omega, \\ \boldsymbol{\mathcal{C}}(\mathbf{q}, \boldsymbol{\nu}) & \text{on} & \Sigma. \end{array} \right. \tag{2.3}$$

The dependence on Ω is thus in the underlying domain and boundary, but also in the possible dependence of the boundary conditions on the normal.

---

[1] The notation $\boldsymbol{\mathcal{B}}$ is used both for the state constraint $\boldsymbol{\mathcal{B}}(\Omega, \mathbf{q})$ (which includes boundary conditions), and for the field equations themselves, $\boldsymbol{\mathcal{B}}(\mathbf{q})$. Which of the two is meant will be clear from the context.

The aim in shape optimization is to find the optimum design, i.e. the one which minimizes the cost functional. Combining all elements, the divertor design problem is stated as

$$\min_{\Omega \in \mathcal{O}, \mathbf{q}} \quad J(\Omega, \mathbf{q}) \quad \text{subject to} \quad \boldsymbol{\mathcal{B}}(\Omega, \mathbf{q}) = 0. \tag{2.4}$$

In this formulation, Ω and $\mathbf{q}$ are considered independent variables, but they are constrained by the state equations. However, it can be assumed that every domain Ω uniquely defines a corresponding plasma state $\mathbf{q} = \mathbf{q}(\Omega)$ through the solution of the state equations. This leads to the equivalent optimization problem

$$\min_{\Omega \in \mathcal{O}} \quad \hat{J}(\Omega) \equiv J(\Omega, \mathbf{q}(\Omega)). \tag{2.5}$$

$\hat{J}(\Omega)$ is called the *reduced* cost functional. By restricting Ω to a set of admissible domains $\mathcal{O}$, further constraints on the design space can be taken into account. These constraints are called control constraints or design constraints. Also additional equality constraints on the states are possible. These are handled in an analogous way to the state equation constraint, see for example Ref. [56], and may be considered a part of $\boldsymbol{\mathcal{B}}(\Omega, \mathbf{q}) = 0$. In the definition of the optimization problems (2.4) and (2.5), there are no inequality constraints on the state variables. These are very difficult to handle theoretically, but have been treated in practice, see for example Ref. [13] and further references therein.

The formulations (2.4) and (2.5) describe in general notation the type of problems that will be treated in this thesis. The rest of this chapter deals with the characterization of solutions to these optimization problems. This will both give insight into the nature of the problems, and lead to efficient solution algorithms. Before proceeding, the Lagrangian formulation of the cost functional is introduced, which in turn requires the variational form of the state equations.

Variational formulation of the state equations

In the context of optimization with PDE constraints, it is convenient to consider the PDE with its boundary conditions (2.3) in variational form,

$$\int_{\Omega} \mathbf{q}^* \cdot \boldsymbol{\mathcal{B}}(\mathbf{q}) \, \mathrm{d}\boldsymbol{\omega} + \int_{\Sigma} \mathbf{p}^* \cdot \boldsymbol{\mathcal{C}}(\mathbf{q}, \boldsymbol{\nu}) \, \mathrm{d}\sigma = 0, \tag{2.6}$$

for all test functions $\mathbf{q}^*$ and $\mathbf{p}^*$ from appropriate Hilbert spaces. In accordance with the state equations, $\mathbf{q}^*$ is a vector of functions with as many components as there are PDEs in $\boldsymbol{\mathcal{B}}(\mathbf{q}) = 0$. A similar argument holds for $\mathbf{p}^*$. Notation may be simplified by introducing the standard L^2 inner products

$$\begin{aligned}(\mathbf{a},\mathbf{b})_\Omega &= \int_\Omega \mathbf{a}\cdot\mathbf{b}\,\mathrm{d}\boldsymbol{\omega},\\ (\mathbf{a},\mathbf{b})_\Sigma &= \int_\Sigma \mathbf{a}\cdot\mathbf{b}\,\mathrm{d}\sigma,\end{aligned}$$

and defining

$$(\mathbf{q}^*,\boldsymbol{\mathcal{B}}(\Omega,\mathbf{q})) \equiv (\mathbf{q}^*,\boldsymbol{\mathcal{B}}(\mathbf{q}))_\Omega + (\mathbf{p}^*,\boldsymbol{\mathcal{C}}(\mathbf{q},\boldsymbol{\nu}))_\Sigma\,. \tag{2.7}$$

Again, the same notation $\mathbf{q}^*$ is used to indicate either the complete set of test functions on domain and boundary, $(\mathbf{q}^*,\mathbf{p}^*)^T$, or the test functions on Ω only. Thus, the state equations (2.3) are equivalent to

$$(\mathbf{q}^*,\boldsymbol{\mathcal{B}}(\Omega,\mathbf{q})) = 0 \tag{2.8}$$

for all $\mathbf{q}^*$.

Saddle point formulation of the cost functional

Consider the Lagrangian function

$$L(\Omega,\mathbf{q},\mathbf{q}^*) = J(\Omega,\mathbf{q}) + (\mathbf{q}^*,\boldsymbol{\mathcal{B}}(\Omega,\mathbf{q}))\,, \tag{2.9}$$

which adds the state equations to the cost functional through an inner product with Lagrange multipliers $\mathbf{q}^*$. In the Lagrangian $L(\Omega,\mathbf{q},\mathbf{q}^*)$, Ω, $\mathbf{q}$ and $\mathbf{q}^*$ are independent variables. Although the symbol $\mathbf{q}$ is used, it is understood that this is just a variable which may take any value from an appropriate function space, and is only equal to the state variables if $\mathbf{q}=\mathbf{q}(\Omega)$ such that $\boldsymbol{\mathcal{B}}(\Omega,\mathbf{q}(\Omega))=0$. Below, it will turn out that choosing $\mathbf{q}^*=\mathbf{q}^*(\Omega)$ as the solution to a dual or adjoint problem will significantly simplify the computation of derivatives of the cost functional. Therefore, the variables $\mathbf{q}^*$ will also be called dual variables or *adjoint* variables, but strictly speaking, this is again only the case if they satisfy the adjoint equations.

It can be observed that [29]

$$J(\Omega,\mathbf{q}(\Omega)) = \min_{\mathbf{q}}\sup_{\mathbf{q}^*} L(\Omega,\mathbf{q},\mathbf{q}^*), \tag{2.10}$$

because

$$\sup_{\mathbf{q}^*} L(\Omega,\mathbf{q},\mathbf{q}^*) = \begin{cases} J(\Omega,\mathbf{q}(\Omega)) & \text{if} \quad \mathbf{q}=\mathbf{q}(\Omega),\\ +\infty & \text{if} \quad \mathbf{q}\neq\mathbf{q}(\Omega).\end{cases}$$

$J(\Omega, \mathbf{q}(\Omega))$ is located at a saddle point of the Lagrangian. This saddle point is characterized by the equations of a stationary point, i.e. by setting the (Gateau-) derivatives with respect to $\mathbf{q}$ and $\mathbf{q}^*$ equal to zero,

$$\left\{ \begin{array}{lllll} 0 & = & \nabla_{\mathbf{q}^*} L(\Omega, \mathbf{q}, \mathbf{q}^*) & = & \boldsymbol{\mathcal{B}}(\Omega, \mathbf{q}), \\ 0 & = & \nabla_{\mathbf{q}} L(\Omega, \mathbf{q}, \mathbf{q}^*) & = & \nabla_{\mathbf{q}} J(\Omega, \mathbf{q}) + \boldsymbol{\mathcal{B}}^*_{\mathbf{q}}(\Omega, \mathbf{q})\mathbf{q}^*. \end{array} \right. \tag{2.11}$$

A variable in subscript means differentiation with respect to that variable. The first equation in (2.11) is straightforward to derive. The second equation — the adjoint equation — follows from first linearizing the state equation with respect to $\mathbf{q}$, and then using

$$(\mathbf{q}^*, \boldsymbol{\mathcal{B}}_{\mathbf{q}}(\Omega, \mathbf{q})\delta\mathbf{q}) = \left(\boldsymbol{\mathcal{B}}^*_{\mathbf{q}}(\Omega, \mathbf{q})\mathbf{q}^*, \delta\mathbf{q}\right),$$

where the (Hilbert space) adjoint operator A^* of a linear operator A is defined through

$$(y, Ax) = (A^* y, x).$$

In later chapters, the edge plasma model and the corresponding adjoint equations will be elaborated. Now, it is only remarked that the saddle point is found by solving the state and adjoint equations (2.11).

2.2 Definition of the Design Space

The control variable in shape optimization problems is a special one, since it is the domain on which the problem is defined, and not a parameter or source term in the state equations or boundary conditions. For many optimization algorithms, information is required on how the cost functional changes with the design variable, i.e. shape sensitivities have to be provided. Therefore, careful attention is paid to two closely linked issues: the description of the geometry, and the computation of sensitivities with respect to the geometry. Over the past decades, optimal shape design has been applied to a variety of problems, mainly in aerodynamic design. Depending on the specific application, different methods for the description of the geometry have been developed. Here, some popular options are briefly reviewed. For convenience, they are roughly divided in three groups, although differences between them are sometimes only subtle. Advantages and disadvantages of the approaches in terms of divertor design will be highlighted.

Shape parametrization

A very intuitive way of describing the shape of the domain is by choosing a suitable parametrization of the domain itself or its boundary. In this case,

$\Omega = \Omega(\phi)$ with ϕ a vector of design parameters. Typically, these parameters are the coefficients in a certain representation of the boundary. For example, Hicks-Henne functions [58] are very popular for airfoil design. Alternatives include the use of B-splines [1] or Bézier curves [31, 32, 126]. These methods have in common that the regularity of the shapes can be guaranteed. Furthermore, by a careful choice of the parametrization, parameters can be directly related to specific geometrical properties of the design such as lengths, tilt angles, inlet cross-sections,... This is illustrated in Fig. 2.1 for a divertor slab model, where it is assumed that effects of magnetic curvature can be neglected in a first approximation [116]. In the figure, the design parameters ϕ_i are tilt angles of the plates and lengths of the divertor legs. Such parametrization can allow an easy formulation of design constraints in terms of the parameters, or even an elimination of constraints by taking them into account directly in the parametrization [126]. On the other hand, finding a good parametrization can be very problem dependent. Moreover, a parametrization will restrict the design space, so that potentially better solutions may be unintentionally removed.

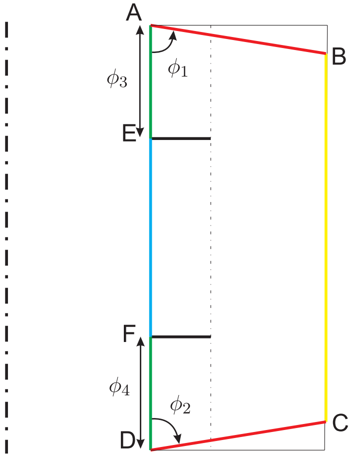


Figure 2.1: Example of divertor geometry parametrization (slab model).

When using a shape parametrization technique, sometimes the relation between grid metrics and design parameters can be written explicitly, and variations of cost functionals such as (2.1) or (2.2) can then be computed directly by taking the derivatives with respect to the design parameters. In realistic design problems, on the other hand, this is usually not possible. In those cases, a hidden difficulty is in the fact that the entire grid generator has to be differentiated. Depending on problem size and complexity of the grid generator algorithm, this may become very challenging. The derivatives could be obtained through finite differences. However, this may consume a large fraction of the benefits of the adjoint and one-shot approaches to be discussed further on. In Ref. [40], the differentiation is performed using AD-software (Algorithmic Differentiation).

In the initial phase of the doctoral research, as well as in Ref. [118], simple parametrizations as illustrated in Fig. 2.1 have been used to validate the numerical implementation of the optimization problem and to prove the potential of automated divertor design. However, in order to add design freedom, attention has shifted to more flexible methods.

Mapping to a reference domain

In a second approach, a coordinate transformation M is performed from the physical domain to a fixed reference domain or (body fitted) curvilinear coordinate system (x, y). As the domain deforms during the design process, the metric coefficients in the transformation change correspondingly. Compared to a parametrization of the shape, a large design freedom is retained: after discretization of the continuous domain, the coordinates of every grid point on the boundary will serve as control variables, which easily leads to a (very) large number of design variables.

In practice, the method is the most valuable if the reference domain is also the computational domain. The governing equations are translated to the (x, y) coordinate system, i.e. the grid system. Usually, structured body fitted grids are used. By definition, the physical boundaries have fixed coordinates in this reference domain, often coinciding with one of the coordinate axes. For example in the work of Jameson, one of the axes is the airfoil [63, 65], or for 3D problems, the wing and the fuselage body coincide with two of the coordinate planes [98]. Generalizations for boundaries not coinciding with coordinate axes can be made [65, 47]. In this case, unstructured grids are often used.

For application to divertor modeling, mapping to a fixed reference domain involves a double coordinate transformation, see Fig. 2.2: first from the cylindrical (R, Z) system to the (curvilinear, orthogonal) poloidal-radial (θ, r) system attached to the magnetic field, followed by a transformation to the geometry fitted grid system (x, y). Due to the highly anisotropic transport in a magnetically confined plasma, the intermediate (θ, r) system is the most convenient to work in. Indeed, the governing equations and boundary conditions are more easily expressed in this system. Furthermore, the transport anisotropy requires the use of field aligned grids to avoid unacceptably large numerical diffusion. As a result, the final transformation to the geometry fitted (x, y) coordinate system is not straightforward, since the divertor and vessel structures are usually not aligned with or perpendicular to the field. Therefore, in general

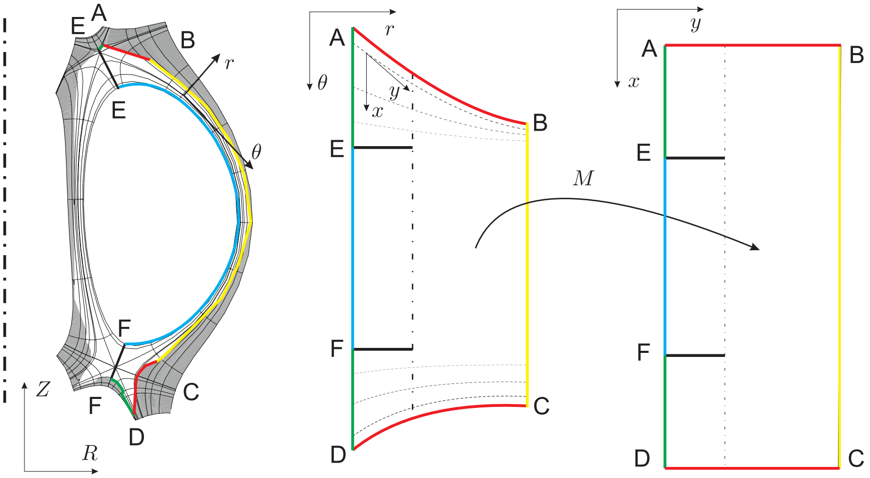


Figure 2.2: Mapping the edge geometry to a reference domain.

a geometry fitted (x, y) coordinate system will be non-orthogonal and not field aligned. The approach to this problem in edge codes as B2 is to work with topologically rectangular grids with quadrilateral cells, where two faces are aligned with the field and the other two are allowed to deviate from the r direction to match the target surface. By keeping the x and θ directions parallel to each other, large numerical diffusion is avoided, while the target surfaces (AB) and (CD) are still resolved due to distortion of radial lines. In practice, the non-orthogonality of the (x, y) system at the targets is often neglected, making the grid behave essentially orthogonal. The effect of this simplification will be studied in Chapter 4. On the other hand, matching the main chamber wall with a topologically rectangular grid would also require distortion of the poloidal coordinate lines, which would induce unacceptable numerical errors. Therefore, the simulations do not extend to the true vessel wall, but only to the last flux surface (BC) that lies completely inside the vessel. Artificial boundary conditions are imposed here. A remedy to this problem is the use of wide grids, see Refs. [6, 24]. The wide grid approach allows to use field aligned cells up to the actual vessel structures by a cut cell approach and by eliminating the restriction to use topologically rectangular grids. With these considerations in mind, and in order to include the wide grid approach from the start, the use of a geometry fitted coordinate system for edge plasma design does not seem appropriate. Therefore, a method is sought which allows to formulate the entire problem in the physically convenient (θ, r) coordinate system, yet has the same design flexibility.

Nonparametric approach through shape sensitivity analysis

A third category of methods which is discussed here, is a nonparametric approach through shape sensitivity analysis. In this very general approach, the problem is described in a fixed coordinate system in which the boundary is allowed to deform freely (with some restrictions on regularity or continuity class of the boundary), as if it were flowing through the physical space with some *design velocity* field $\boldsymbol{\mathcal{V}}$, see Fig. 2.3. Therefore, there is again a large design freedom, only limited by grid resolution.

The nonparametric approach is based on a framework of tangential calculus on the boundary of the domain, which gives clear insight into the nature of the shape sensitivities. Furthermore, the Hadamard-Zolésio structure theorem [29] allows to formulate sensitivities with respect to the shape of the domain in terms of boundary data only. Thus, the sensitivities can be computed at a very low computational cost. This is very useful for the one-shot optimization approach, as it requires frequent evaluation of the sensitivities. Since no reference is made to the underlying grid, any suitable coordinate system can be used to formulate the problem. Specifically, the convenient poloidal-radial system may be used. Furthermore, as long as the boundary data is sufficiently accurate this approach can work with both structured and unstructured grids, and is in particular very promising for use with wide grids. The nonparametric approach is state of the art, and has been applied successfully by several authors for a range of problems [61, 38, 109], including large-scale applications [108].

In the method using a reference domain described above, a change in design will lead to a corresponding change in metric coefficients in the entire domain due to the underlying coordinate transformation. Thus, the resulting expressions for variations with respect to domain shape will also consist of volume integrals, which can be expensive to evaluate. However, it can be argued that for a fixed physical boundary, changes in the coordinate system should not influence the value of the cost functional. Using this notion, the domain expressions for the shape sensitivities can be simplified to boundary expressions, see Ref. [64]. Although a formal transformation to a reference domain is not done in Refs. [89, 43], the approach presented there refers explicitly to the grid coordinate system when variations of shape dependent terms have to be evaluated. Similar arguments as in Ref. [64] are then used to reduce volume expressions to boundary expression for the shape sensitivity. The resulting formulas for shape sensitivities are equivalent to the ones obtained directly with shape sensitivity analysis.

Given its generality, the possibility to work in the (θ, r) coordinate system and the efficient expressions for shape derivatives on the boundary of the domain, it is chosen to work with the nonparametric approach in this thesis. Note that the sensitivity information can easily be translated to a specific parametrization $\boldsymbol{\mathcal{V}}(\phi)$ if required. For example, field aligned grids can be ensured by only using velocity fields with zero radial component. In this case, the design is parametrized by a function $\phi(r)$ which determines the poloidal location of the target, see Fig. 2.3.

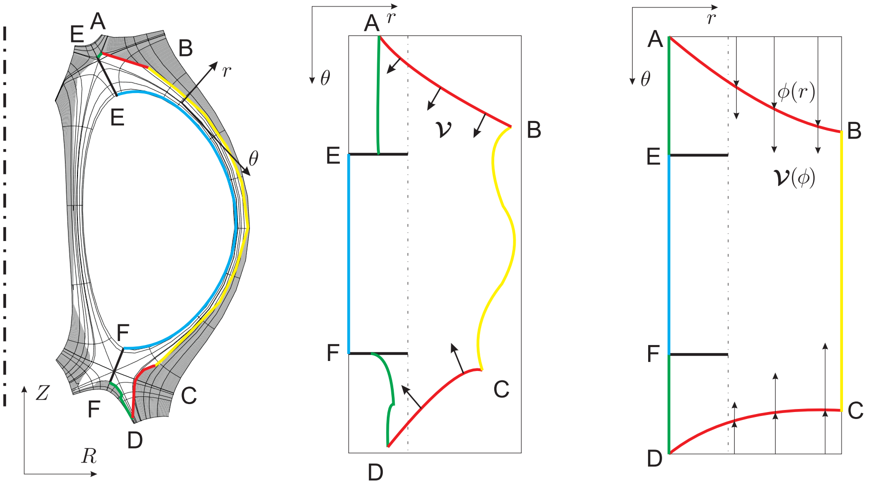


Figure 2.3: Middle: geometry 'flowing' along design velocity field $\boldsymbol{\mathcal{V}}$. Right: a specific choice of velocity field $\boldsymbol{\mathcal{V}}(\phi)$ allows to keep the grid field aligned.

2.3 Basics of Shape Sensitivity Analysis

In this section, some elements of shape sensitivity analysis useful for this thesis are highlighted. Details on this topic can be found in several text books. For a rigorous mathematical treatment, see Refs. [114, 29]. A more formal approach, on which the discussion below is based, can be found in Refs. [52, 19].

2.3.1 The Material Derivative Approach

The ideas underlying shape sensitivity analysis resemble the use of material derivatives in fluid mechanics. The domain Ω is regarded as a continuum which deforms due to changes in design variables, so that at a certain instant $t \geq 0$ in *design time*, it has a configuration Ω_t (Fig. 2.4). At $t = 0$, the initial domain

$\Omega = \Omega_0$ is retrieved. Thus, a family of mappings $\mathbf{T}_t$, parametrized by $t \in [0, \tau]$, of the original domain to the deformed domain can be considered, mapping a point $\boldsymbol{\omega}$ in Ω to $\boldsymbol{\omega}_t$:

$$\boldsymbol{\omega}_t = \mathbf{T}_t(\boldsymbol{\omega}) \quad \text{for} \quad \boldsymbol{\omega} \in \Omega.$$

In order to have physically meaningful transformations, this mapping must be a bijection, and such that Ω is mapped to Ω_t and the boundary Σ is mapped to the deformed boundary $\Sigma_t = \partial\Omega_t$:

$$\Omega_t = \mathbf{T}_t(\Omega) \quad \text{and} \quad \Sigma_t = \mathbf{T}_t(\Sigma).$$

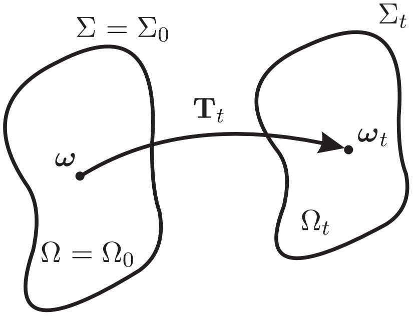


Figure 2.4: Transformation of initial domain Ω to a perturbed domain Ω_t.

Two transformations frequently used are the perturbation of identity and the velocity method. Both can be interpreted in terms of a design velocity field $\boldsymbol{\mathcal{V}}$. For the perturbation of identity, the transformation is

$$\mathbf{T}_t(\boldsymbol{\omega}) = \boldsymbol{\omega} + t\boldsymbol{\mathcal{V}}(\boldsymbol{\omega}), \tag{2.12}$$

while in the velocity method, $\mathbf{T}_t(\boldsymbol{\omega}) = \mathbf{x}(t, \boldsymbol{\omega})$ is the solution to the flow problem

$$\frac{\partial \mathbf{x}}{\partial t} = \boldsymbol{\mathcal{V}}(t, \mathbf{x}), \quad \mathbf{x}(0, \boldsymbol{\omega}) = \boldsymbol{\omega}. \tag{2.13}$$

In contrast to the perturbation of identity, the velocity method allows for non constant design velocity fields in time. By taking a Taylor series expansion of $\mathbf{T}_t(\boldsymbol{\omega})$ around $t = 0$, it can be seen that both transformations are the same up to first order in time, and thus will lead to the same first order shape derivatives (but not necessarily the same higher order derivatives). Since only fist order derivatives are needed in this work, expression (2.12) for the perturbation of identity will be used for convenience.

For a sufficiently smooth function $\mathbf{q}_t(\boldsymbol{\omega}_t) = \mathbf{q}(t, \boldsymbol{\omega} + t\boldsymbol{\mathcal{V}}(\boldsymbol{\omega}))$, the total derivative or (pointwise) material derivative $\dot{\mathbf{q}}$ is found by the chain rule,

$$\dot{\mathbf{q}}(0,\boldsymbol{\omega}) \equiv \frac{\mathrm{d}}{\mathrm{d}t}\mathbf{q}(t,\boldsymbol{\omega}+t\boldsymbol{\mathcal{V}}(\boldsymbol{\omega}))\bigg|_{t=0+} = \frac{\partial}{\partial t}\mathbf{q}(0,\boldsymbol{\omega}) + \boldsymbol{\mathcal{V}}(\boldsymbol{\omega})\cdot\nabla\mathbf{q}(0,\boldsymbol{\omega}).$$

The partial derivative of $\mathbf{q}$ with respect to t is called the (pointwise) shape derivative, and is usually written as $\mathbf{q}'$. For ease of notation, let $\mathbf{q}$ denote both the function $\mathbf{q}(t, \boldsymbol{\omega}_t)$ and its restriction to $t = 0$, $\mathbf{q}(0, \boldsymbol{\omega}) = \mathbf{q}(\boldsymbol{\omega})$, so

$$\dot{\mathbf{q}} = \mathbf{q}' + \boldsymbol{\mathcal{V}}\cdot\nabla\mathbf{q}. \tag{2.14}$$

Since t and $\boldsymbol{\omega}$ are independent variables, the shape derivative satisfies

$$\left(\frac{\partial\mathbf{q}}{\partial\boldsymbol{\omega}}\right)' = \frac{\partial\mathbf{q}'}{\partial\boldsymbol{\omega}}.$$

The shape derivative of a general shape functional $J(\Omega)$ is defined as[2]

$$\dot{J} \equiv \dot{J}(\Omega;\boldsymbol{\mathcal{V}}) = \lim_{t\to 0+}\frac{J(\Omega_t)-J(\Omega)}{t} = \frac{\mathrm{d}}{\mathrm{d}t}J(\Omega_t)\bigg|_{t=0+}, \tag{2.15}$$

if the limit exists for all $\boldsymbol{\mathcal{V}}$ and the mapping $\boldsymbol{\mathcal{V}} \mapsto \dot{J}(\Omega;\boldsymbol{\mathcal{V}})$ is linear and continuous [29]. Thus, the shape derivative is a directional derivative for a perturbation in direction $\boldsymbol{\mathcal{V}}$, which is stressed by keeping $\boldsymbol{\mathcal{V}}$ in the notation $\dot{J}(\Omega;\boldsymbol{\mathcal{V}})$. According to the Hadamard-Zolésio structure theorem [29], under certain assumptions this derivative can always be written as

$$\dot{J} = \int_{\Sigma}\mathcal{G}\,\boldsymbol{\mathcal{V}}\cdot\boldsymbol{\nu}\,\mathrm{d}\sigma. \tag{2.16}$$

The scalar distribution $\mathcal{G}$ will be called the shape gradient. This result is very powerful, because only boundary integrals have to be evaluated to obtain the shape derivative. In general, these are cheap to compute compared to domain integrals. Furthermore, the expression only depends on the normal component of the design velocity on the boundary. While this is an intuitive result (deformations tangential to the boundary do not change the domain shape), it also means that no assumptions have to be made about how the interior of the domain deforms. Therefore, the Jacobian of the grid generator is not required, a big computational advantage. The final result of the structure theorem is a prescription to modify directly the boundary of the domain that can be used with any type of grid, both structured and unstructured, and in particular with wide grids.

---

[2] The term *shape derivative* is used both for the shape derivative of a functional, Eq. (2.15), and for the shape derivative of a function, $\mathbf{q}'$ in Eq. (2.14). Which of the two is referred to should be clear from the context.

2.3.2 Shape Derivatives of Domain and Boundary Integrals

In this section, shape derivatives of some frequently appearing functionals are discussed. These shape derivatives will then be used throughout the thesis. However, because the derivations of some of these expressions are quite technical, they have been moved to Appendix A to improve the readability of the text. The interested reader may also refer to this appendix for an overview of some further useful expressions regarding the transformation $\mathbf{T}_t$ and derivatives of the surface normal.

Shape derivative of domain integral

A first type of functional which is frequently encountered is an integral across the domain,

$$J(\Omega_t) = \int_{\Omega_t} f\left(t, \boldsymbol{\omega}_t\right) \,\mathrm{d}\boldsymbol{\omega}_t. \tag{2.17}$$

This type of integral arises when a cost functional is defined across the whole domain, as is Eq. (2.1), but also in the variational formulation of the state equations, Eq. (2.6). In order to compute the shape derivative of this functional, the transformation $\mathbf{T}_t$ is first used to bring the integral back to the unperturbed domain Ω, and then the order of differentiation and integration can be switched:

$$\begin{aligned} \dot{J}(\Omega; \boldsymbol{\mathcal{V}}) &= \frac{\mathrm{d}}{\mathrm{d}t}\left(\int_{\Omega_t} f\left(t, \boldsymbol{\omega}_t\right) \,\mathrm{d}\boldsymbol{\omega}_t\right)\Bigg|_{t=0+} \\ &= \int_{\Omega} \frac{\mathrm{d}}{\mathrm{d}t}\left(f\left(t, \boldsymbol{\omega} + t\boldsymbol{\mathcal{V}}(\boldsymbol{\omega})\right) \left|D\mathbf{T}_t(\boldsymbol{\omega})\right|\right)\Bigg|_{t=0+} \mathrm{d}\boldsymbol{\omega}. \end{aligned}$$

D is the Jacobian operator, and $|D\mathbf{T}_t(\boldsymbol{\omega})|$ the Jacobian of the transformation, i.e. the determinant of the Jacobian matrix. Then, using the expressions (A.19) from Appendix A and (2.14), and applying the divergence theorem gives

$$\begin{aligned} \dot{J}(\Omega; \boldsymbol{\mathcal{V}}) &= \int_{\Omega} \left(\dot{f} + f\, \nabla \cdot \boldsymbol{\mathcal{V}}\right) \mathrm{d}\boldsymbol{\omega} \\ &= \int_{\Omega} f' \,\mathrm{d}\boldsymbol{\omega} + \int_{\Sigma} f\, \boldsymbol{\mathcal{V}} \cdot \boldsymbol{\nu} \,\mathrm{d}\sigma. \end{aligned} \tag{2.18}$$

In the context of fluid mechanics, this result is well known as the Reynolds transport theorem. From the formula above, it is seen that there are two equivalent expressions for the shape derivative of a domain integral. The last equality involves a boundary integral, and can be brought back to the Hadamard form (2.16). This is obvious in case f does not depend explicitly on t (i.e. if $f' = 0$). If f does depend on t, further elaboration is needed, see Section 2.3.3. However, the shape derivative in terms of boundary integrals requires higher smoothness of the solution than the first expression for the shape derivative, in which only a domain integral appears. Indeed, in order to apply the divergence theorem, f must be continuously differentiable on the domain. This smoothness can typically not be achieved with (low order) finite element solutions, and then the domain expression must be used [52]. This also means that an extension of the design velocity field $\boldsymbol{\mathcal{V}}$ has to be assumed in the entire domain. Usually, $\boldsymbol{\mathcal{V}}$ will then be linked to grid deformation. With the finite volume approach used in this thesis, the smoothness will be sufficient to use the boundary expression.

In order to simplify notation in the remainder of the text, shape derivatives of functionals will be split in two parts. The integral involving the shape derivative of the integrand will be denoted with a prime, J', while the part of the derivative which is in Hadamard form will be written out explicitly:

$$\dot{J} = J' + \int_{\Sigma} f\, \boldsymbol{\mathcal{V}} \cdot \boldsymbol{\nu}\, \mathrm{d}\sigma, \quad \text{with} \quad J' = \int_{\Omega} f'\, \mathrm{d}\boldsymbol{\omega}. \tag{2.19}$$

In Section 2.3.3 it will be shown that under a state equation constraint, all terms of the form J' in the derivative of the cost functional cancel if the state and adjoint equations are satisfied, so the Hadamard form of the shape derivative will be retrieved.

Shape derivatives of boundary integrals

Also several types of functionals defined on the boundary of the domain have to be dealt with. Often, these functionals depend on the surface normal $\boldsymbol{\nu}$. Typical examples are

$$J_1(\Omega_t) = \int_{\Sigma_t} g\left(t, \boldsymbol{\omega}_t\right) \mathrm{d}\sigma_t, \tag{2.20}$$

$$J_2(\Omega_t) = \int_{\Sigma_t} \mathbf{g}\left(t, \boldsymbol{\omega}_t\right) \cdot \boldsymbol{\nu}_t\, \mathrm{d}\sigma_t, \tag{2.21}$$

$$J_3(\Omega_t) = \int_{\Sigma_t} h\left(\mathbf{q}(t, \boldsymbol{\omega}_t), \boldsymbol{\nu}_t\right) \mathrm{d}\sigma_t. \tag{2.22}$$

It is noted that functionals J_2 and J_3 are special instances of J_1, but because they appear so frequently it is useful to isolate the dependence on the surface normal. Cost functionals defined on the boundary are often of type J_3. In the variational form of the state equations, Dirichlet boundary conditions will lead to integrals of type J_1. Neumann conditions, on the other hand, can be formulated as integrals of type J_2. The derivation of the shape derivatives of these functionals is quite technical, and is moved to Appendix A. Details can also be found in Refs. [52, 19, 29], and in particular for the third functional in Refs. [108, 110]. Here, only the results are stated:

$$\dot{J}_1(\Omega;\boldsymbol{\mathcal{V}}) = \int_\Sigma g' \,\mathrm{d}\sigma + \int_\Sigma (\boldsymbol{\nu}\cdot\nabla g + g\kappa)\,\boldsymbol{\mathcal{V}}\cdot\boldsymbol{\nu}\,\mathrm{d}\sigma, \tag{2.23}$$

$$\dot{J}_2(\Omega;\boldsymbol{\mathcal{V}}) = \int_\Sigma \mathbf{g}'\cdot\boldsymbol{\nu}\,\mathrm{d}\sigma + \int_\Sigma (\nabla\cdot\mathbf{g})\,\boldsymbol{\mathcal{V}}\cdot\boldsymbol{\nu}\,\mathrm{d}\sigma, \tag{2.24}$$

$$\dot{J}_3(\Omega;\boldsymbol{\mathcal{V}}) = \int_\Sigma h_{\mathbf{q}}\mathbf{q}'\,\mathrm{d}\sigma + \int_\Sigma \Big(h_{\mathbf{q}}D\mathbf{q}\,\boldsymbol{\nu} + \kappa\,(h - h_{\boldsymbol{\nu}}\boldsymbol{\nu}) + \nabla_\Sigma\cdot(h_{\boldsymbol{\nu}})^T\Big)\,\boldsymbol{\mathcal{V}}\cdot\boldsymbol{\nu}\,\mathrm{d}\sigma. \tag{2.25}$$

$\kappa = \nabla_\Sigma\cdot\boldsymbol{\nu}$ is the additive curvature, i.e. the sum of the principal curvatures of the boundary, and ∇_Σ is the tangential gradient operator, see Appendix A. Also here, the integrals involving the shape derivative of the integrand are isolated by defining J_1', J_2' and J_3' such that

$$\begin{aligned}
\dot{J}_1(\Omega;\boldsymbol{\mathcal{V}}) &= J_1' + \int_\Sigma (\boldsymbol{\nu}\cdot\nabla g + g\kappa)\,\boldsymbol{\mathcal{V}}\cdot\boldsymbol{\nu}\,\mathrm{d}\sigma,\\
\dot{J}_2(\Omega;\boldsymbol{\mathcal{V}}) &= J_2' + \int_\Sigma (\nabla\cdot\mathbf{g})\,\boldsymbol{\mathcal{V}}\cdot\boldsymbol{\nu}\,\mathrm{d}\sigma,\\
\dot{J}_3(\Omega;\boldsymbol{\mathcal{V}}) &= J_3' + \int_\Sigma \Big(h_{\mathbf{q}}D\mathbf{q}\,\boldsymbol{\nu} + \kappa\,(h - h_{\boldsymbol{\nu}}\boldsymbol{\nu}) + \nabla_\Sigma\cdot(h_{\boldsymbol{\nu}})^T\Big)\,\boldsymbol{\mathcal{V}}\cdot\boldsymbol{\nu}\,\mathrm{d}\sigma.
\end{aligned}$$

None of the shape derivatives of domain and boundary integrals given above satisfy the Hadamard form (2.16) yet. The reason is the dependence of the integrands on t, leading to the shape derivatives f', g', $\mathbf{g}'$, and $h_{\mathbf{q}}\mathbf{q}'$. This dependence is detailed in the next section for the case of integrals depending on state variables.

2.3.3 Shape Derivatives of Cost Functionals Depending on State Variables

In case of a state constrained shape optimization problem, the dependence on t is directly or indirectly related to the state variables, which have to satisfy an equation

$$\boldsymbol{\mathcal{B}}(\Omega_t, \mathbf{q}_t) = 0 \tag{2.26}$$

on the perturbed domain. Again, there are several basic approaches to take this constraint into account.

- A first method uses the state constraint (2.26) to consider the state variable as a dependent variable $\mathbf{q}_t = \mathbf{q}_t(\Omega_t)$. In this case, the dependence of a cost functional on t can be made explicit by writing for example $f(t, \boldsymbol{\omega}_t) = F(\mathbf{q}_t(\boldsymbol{\omega}_t)) = F(\mathbf{q}(t, \boldsymbol{\omega}_t))$ in the domain integral (2.17). Then, the chain rule is applied to the reduced cost functional $\hat{J}(\Omega_t) = J(\Omega_t, \mathbf{q}_t(\Omega_t))$. The chain rule relates f' to the shape derivatives of the state variables, $f' = F_{\mathbf{q}}\mathbf{q}'$. By differentiating (2.26) with respect to the shape, a set of (linearized) equations and boundary conditions for $\mathbf{q}'$ can be derived. By constructing an appropriate adjoint problem (on the unperturbed domain), the field integrals can finally be eliminated. This approach is followed in Refs. [114, 52, 108, 92].
- In control theory, it is customary to regard the control and state variables as independent, and enforce the state equation through the definition of a Lagrangian. As discussed at the end of Section 2.1, the cost functional is then located at a saddle point of the Lagrangian. The dependence on t is through a specific parametrization of the Lagrangian. This approach is followed in Refs. [29, 38].
- In Refs. [61, 69], the state variable is again considered as a dependent variable, but the computation of shape derivatives of the state variables is avoided by a method of mappings which brings the state equation on perturbed domain back to the unperturbed domain.

On a technical level, the last two approaches avoid the need to compute the shape derivative of the state variables, which means the results are also valid for more irregular problems [29, 61, 69]. These shape derivatives can be a delicate issue, since the derivative is with respect to the domain on which the function is defined. In any case, the same final expressions for shape derivatives are obtained with all three methods. From a practical point of view, the control theory approach with a Lagrangian leads to an elegant and convenient formal framework to derive the adjoint equations. Furthermore, the resulting optimality conditions give clear insight into the structure of the optimization problem, and the method naturally extends to control variables other than the shape of the domain. Therefore, it is chosen to work with the Lagrangian approach here.

Shape derivative of the saddle point

With the cost functional formulated as the saddle point of a Lagrangian, Eq. (2.10), its shape derivative can be found by differentiating this saddle point with respect to design time t. For a theoretical development, see Ref. [29]. On the perturbed domain, the saddle point is

$$J(\Omega_t, \mathbf{q}(\Omega_t)) = \min_{\mathbf{q}_t} \sup_{\mathbf{q}_t^*} L(\Omega_t, \mathbf{q}_t, \mathbf{q}_t^*).$$

Since the Lagrangian is constructed as a sum of domain and boundary integrals, Eq. (2.9), its shape derivative will be found by combining expressions from the previous section. However, in order to obtain the Hadamard form of the shape derivative, special care is needed regarding the time dependence of the integrands. In the function space parametrization method [29], this dependence is taken care of by using the inverse transformation $\mathbf{T}_t^{-1}$ to associate with every function $\mathbf{q}_t(\boldsymbol{\omega}_t)$ and $\mathbf{q}_t^*(\boldsymbol{\omega}_t)$ defined on the perturbed domain Ω_t a corresponding element $\mathbf{q}(\boldsymbol{\omega})$, respectively $\mathbf{q}^*(\boldsymbol{\omega})$, on the unperturbed domain which is transported along $\boldsymbol{\mathcal{V}}$:

$$\begin{aligned}
\mathbf{q}_t(\boldsymbol{\omega}_t) &= \mathbf{q}(\mathbf{T}_t^{-1}(\boldsymbol{\omega}_t)) &&= \mathbf{q} \circ \mathbf{T}_t^{-1}(\boldsymbol{\omega}_t), \\
\mathbf{q}_t^*(\boldsymbol{\omega}_t) &= \mathbf{q}^*(\mathbf{T}_t^{-1}(\boldsymbol{\omega}_t)) &&= \mathbf{q}^* \circ \mathbf{T}_t^{-1}(\boldsymbol{\omega}_t).
\end{aligned}$$

This amounts to a specific parametrization of the Lagrangian, but the value of the saddle point is not changed [29]. The parametrization directly implies that the material derivative of $\mathbf{q}_t(\boldsymbol{\omega}_t)$ is zero, because $\mathbf{q}_t(\boldsymbol{\omega}_t) = \mathbf{q}(t, \boldsymbol{\omega}_t) = \mathbf{q}(0, \boldsymbol{\omega}) = \mathbf{q}(\boldsymbol{\omega})$. Thus, by Eq. (2.14), the shape derivative is

$$\mathbf{q}' = -\boldsymbol{\mathcal{V}} \cdot \nabla \mathbf{q}. \tag{2.27}$$

The same argument holds for $\mathbf{q}^*$, so

$$(\mathbf{q}^*)' = -\boldsymbol{\mathcal{V}} \cdot \nabla \mathbf{q}^*. \tag{2.28}$$

The parametrization has important consequences on the interpretation of $\mathbf{q}_t(\Omega_t)$ as a solution to $\boldsymbol{\mathcal{B}}(\Omega_t, \mathbf{q}_t)$. Rather than seeing $\mathbf{q}_t(\Omega_t)(\boldsymbol{\omega}_t)$ as an element $\mathbf{q}(\Omega)(t, \boldsymbol{\omega}_t)$ which *deforms* in design time due to the change in domain, now for every t, $\mathbf{q}_t(\Omega_t)(\boldsymbol{\omega}_t)$ corresponds to *another* element on the original domain, $\mathbf{q}(\Omega_t)(\boldsymbol{\omega})$which is *transported* along $\boldsymbol{\mathcal{V}}$. This also explains why the shape derivative of the state variables is not needed in the formulation of the optimality conditions. What is required now, is that for any sequence $\{t_n : 0 < t_n \leq \tau\}$ with $t_n \to 0$ there exists a subsequence $\{t_{n_k}\}$ such that $\mathbf{q}(\Omega_{t_{n_k}}) \to \mathbf{q}(\Omega)$.

Returning to the saddle point, the min-sup problem is now formulated as

$$\begin{aligned} J(\Omega_t, \mathbf{q}(\Omega_t)) &= \min_{\mathbf{q}} \sup_{\mathbf{q}^*} L(\mathbf{T}_t(\Omega), \mathbf{q} \circ \mathbf{T}_t^{-1}, \mathbf{q}^* \circ \mathbf{T}_t^{-1}) \\ &\equiv \min_{\mathbf{q}} \sup_{\mathbf{q}^*} \tilde{L}(t, \mathbf{q}, \mathbf{q}^*). \end{aligned}$$

The shape derivative of the cost functional is found by differentiating the saddle point of the Lagrangian with respect to the parameter t [29],

$$\begin{aligned} \dot{J} = \frac{\mathrm{d}}{\mathrm{d}t} J(\Omega_t, \mathbf{q}(\Omega_t))\Big|_{t=0+} &= \min_{\mathbf{q}} \sup_{\mathbf{q}^*} \frac{\partial}{\partial t} \tilde{L}(t, \mathbf{q}, \mathbf{q}^*)\Big|_{t=0+} \\ &= \frac{\partial}{\partial t} \tilde{L}(0, \mathbf{q}(\Omega), \mathbf{q}^*(\Omega)), \end{aligned} \tag{2.29}$$

where $\mathbf{q}(\Omega)$ and $\mathbf{q}^*(\Omega)$ solve the saddle point equations (2.11).

Hadamard form of the shape derivative

Expression (2.29) can now be elaborated to reduce the shape derivative of a general cost functional to the Hadamard form. To this end, it is important to examine first in detail the meaning of $\partial_t \tilde{L}(0, \mathbf{q}, \mathbf{q}^*)$. This notation stresses that the partial derivative of the Lagrangian with respect to t has to be evaluated at $t = 0$. As for all domain and boundary integrals discussed at the beginning of this section, the partial derivative with respect to t involves two types of terms. First, there are terms which involve the partial derivative of the integrand with respect to t. In practice, the Lagrangian is usually available in the form $L(\mathbf{T}_t(\Omega), \mathbf{q} \circ \mathbf{T}_t^{-1}, \mathbf{q}^* \circ \mathbf{T}_t^{-1})$, so these terms are

$$L' = L_{\mathbf{q}}(\Omega, \mathbf{q}, \mathbf{q}^*)\mathbf{q}' + L_{\mathbf{q}^*}(\Omega, \mathbf{q}, \mathbf{q}^*)\,(\mathbf{q}^*)',$$

where the shape derivatives $\mathbf{q}'$ and $(\mathbf{q}^*)'$ of $\mathbf{q} \circ \mathbf{T}_t^{-1}$ and $\mathbf{q}^* \circ \mathbf{T}_t^{-1}$ are given by (2.27) and (2.28). Further elaboration using Eq. (2.9) gives

$$\begin{aligned} L' &= (\nabla_{\mathbf{q}} J(\Omega, \mathbf{q}), \mathbf{q}') + (\mathbf{q}^*, \boldsymbol{\mathcal{B}}_{\mathbf{q}}(\Omega, \mathbf{q})\mathbf{q}') + ((\mathbf{q}^*)', \boldsymbol{\mathcal{B}}(\Omega, \mathbf{q})) \\ &= \left(\nabla_{\mathbf{q}} J(\Omega, \mathbf{q}) + \boldsymbol{\mathcal{B}}_{\mathbf{q}}^*(\Omega, \mathbf{q})\mathbf{q}^*, \mathbf{q}'\right) + ((\mathbf{q}^*)', \boldsymbol{\mathcal{B}}(\Omega, \mathbf{q})) \\ &= 0. \end{aligned} \tag{2.30}$$

These terms are zero because at $t = 0$, $\mathbf{q} = \mathbf{q}(\Omega)$ and $\mathbf{q}^* = \mathbf{q}^*(\Omega)$ solve the state and adjoint equations (2.11). The other terms in the shape derivative of the Lagrangian are in Hadamard form, so the desired result is obtained. Further specification of the shape derivative requires knowledge about the cost functional and the state equations, and is postponed to later chapters.

2.4 Optimality Conditions

2.4.1 First Order Optimality Conditions

Based on the results of the previous sections, it is now possible to identify conditions which characterize solutions $\hat{\Omega}$ and $\hat{\mathbf{q}} = \mathbf{q}(\hat{\Omega})$ to optimization problems of type (2.4) or (2.5), and the corresponding adjoint variables $\hat{\mathbf{q}}^* = \mathbf{q}^*(\hat{\Omega})$.

First, consider the reduced problem (2.5), in which the difficulty stemming from the underlying PDE is hidden. Then, standard optimization theory gives the necessary first order optimality condition

$$\dot{\hat{J}}(\hat{\Omega}; \boldsymbol{\mathcal{V}}) \geq 0 \tag{2.31}$$

for all feasible directions $\boldsymbol{\mathcal{V}}$ at the solution $\hat{\Omega}$. For a proof, see for example Refs. [87, 13]. The interpretation of this condition is shown in Fig. 2.5. For an optimization problem without design constraints, a solution is located at a local minimum of the objective functional, so the derivative must be zero here. If the design space is constrained, it is possible that the minimum is located on the boundary of the constraints. In this case, the value of the objective functional must not decrease for all directions $\boldsymbol{\mathcal{V}}$ which satisfy the design constraints, otherwise $\hat{\Omega}$ is not an optimal solution.

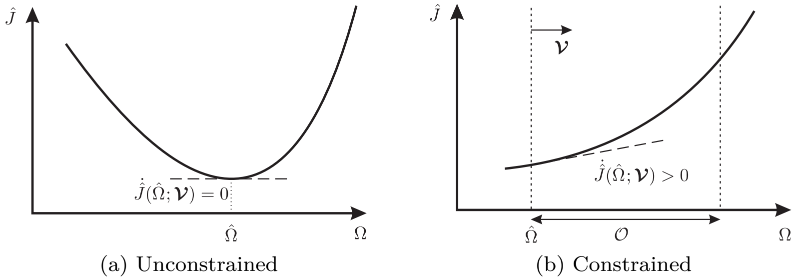
In view of the structure theorem, condition (2.31) means

$$\dot{\hat{J}}(\hat{\Omega}; \boldsymbol{\mathcal{V}}) = \int_{\Sigma} \mathcal{G}\, \boldsymbol{\mathcal{V}} \cdot \boldsymbol{\nu} \, \mathrm{d}\sigma \geq 0.$$

If there are no constraints on the control variable, this relation must hold for all possible $\boldsymbol{\mathcal{V}}$, so at the optimum, $\mathcal{G} = 0$.

The reduced framework does not directly provide a method for efficient computation of the shape derivative. Indeed, because $\hat{J}(\Omega) = J(\Omega, \mathbf{q}(\Omega))$, every evaluation of the cost functional requires a solution of the state equations. Similarly, at least one additional solution of the state equations is required for each design variable if derivative information is obtained with a forward approach, for example using finite differences:

$$\dot{\hat{J}}(\Omega; \boldsymbol{\mathcal{V}}) \approx \frac{\hat{J}(\Omega_t) - \hat{J}(\Omega)}{t}, \tag{2.32}$$



(a) Unconstrained (b) Constrained

Figure 2.5: Graphical representation of optimal solution.

with t sufficiently small. If the design space is discretized using N boundary nodes, the expression above has to be evaluated for N linearly independent velocity fields $\boldsymbol{\mathcal{V}}$, meaning a total of $1+N$ flow solutions are required to obtain the shape derivative. For shape optimization problems, N is usually large. Furthermore, solving the state equations is computationally expensive, so this cost becomes prohibitive. Last but not least, computing a gradient with finite differences may be inaccurate due to numerical errors and the difficult choice of a good perturbation size t.

On the other hand, using the results of the previous section, the shape derivative of the cost functional is given by Eq. (2.29),

$$\dot{\hat{J}}(\Omega;\boldsymbol{\mathcal{V}}) = \dot{J}(\Omega,\mathbf{q}(\Omega);\boldsymbol{\mathcal{V}}) = \frac{\partial}{\partial t}\tilde{L}(0,\mathbf{q}(\Omega),\mathbf{q}^*(\Omega)), \tag{2.33}$$

where $\mathbf{q}(\Omega)$ and $\mathbf{q}^*(\Omega)$ are the solutions to Eqs. (2.11),

$$0 = \nabla_{\mathbf{q}^*}L(\Omega,\mathbf{q},\mathbf{q}^*) = \boldsymbol{\mathcal{B}}(\Omega,\mathbf{q}), \tag{2.34}$$

$$0 = \nabla_{\mathbf{q}}\ L(\Omega,\mathbf{q},\mathbf{q}^*) = \nabla_{\mathbf{q}}J(\Omega,\mathbf{q}) + \boldsymbol{\mathcal{B}}^*_{\mathbf{q}}(\Omega,\mathbf{q})\mathbf{q}^*. \tag{2.35}$$

Thus, by solving first the state equations for $\mathbf{q}(\Omega)$, followed by the adjoint equations for $\mathbf{q}^*(\Omega)$, the shape derivative of the cost functional can be computed at a cost of roughly 2 flow simulations. It must be remarked that a new set of adjoint equations has to be solved for every independent cost functional or *observation* of the system, because the adjoint equations depend on the cost functional. The trade-off between forward or adjoint sensitivity analysis thus depends on the number of control variables N compared to the number of observations N_o. In general, the cost of forward sensitivity analysis is $1+N$ flow simulations, and the cost of adjoint analysis is $1+N_o$ simulations. If $N > N_o$, it pays off to use the adjoint technique. Otherwise, a forward approach will be more efficient. In that case, it should also be considered to solve a linearized system of state equations to compute flow perturbations rather than using finite differences in order to improve numerical accuracy.

Summarizing, the following set of conditions characterize an optimal design for problems (2.4) and (2.5):

$$\left\{\begin{array}{lcll} 0 & = & \nabla_{\mathbf{q}^*} L(\hat{\Omega}, \hat{\mathbf{q}}, \hat{\mathbf{q}}^*) & \text{state equations,} \\ 0 & = & \nabla_{\mathbf{q}}\ L(\hat{\Omega}, \hat{\mathbf{q}}, \hat{\mathbf{q}}^*) & \text{adjoint equations,} \\ 0 & \leq & \Big(\nabla_\Omega L(\hat{\Omega}, \hat{\mathbf{q}}, \hat{\mathbf{q}}^*), \boldsymbol{\mathcal{V}} \cdot \boldsymbol{\nu}\Big) & \text{design equation.} \end{array}\right. \tag{2.36}$$

The notation ∇_Ω has been introduced to indicate formally the structure of the problem, and its similarity to optimal control of PDEs where the control variable is typically a parameter in the PDE, and not the domain on which it is defined. It is understood that $\nabla_\Omega L$ refers to Hadamard shape gradient $\mathcal{G}$ resulting from $\partial_t \tilde{L}$.

2.4.2 Design Updates and Gradient Smoothing

In order to find a design which satisfies the optimality conditions (2.36), an iterative procedure can be used to gradually update the design from some initial configuration to the optimal configuration. The update direction is based on gradient information. However, it is a well known problem in shape design that if individual boundary nodes are the control variables, the shape gradient $\mathcal{G}$ itself may have insufficient smoothness to guarantee the regularity of the boundary [63, 82]. Therefore, oscillations in the shape may appear if no special care is taken.

This problem can be avoided by using a smoothed gradient $\tilde{\mathcal{G}}$ in the design update. In the literature, the domain is often deformed in the direction normal to the boundary by updates of the form $\boldsymbol{\mathcal{V}} = -\alpha \tilde{\mathcal{G}} \boldsymbol{\nu}$ with α the step length. The smoothed gradient is obtained by

$$\tilde{\mathcal{G}} - \epsilon \Delta_\Sigma \tilde{\mathcal{G}} = \mathcal{G}. \tag{2.37}$$

Δ_Σ is the Laplace-Beltrami operator, i.e. the Laplacian acting on the boundary of the domain. ϵ is a numerical parameter which can be used to tune the smoothing. This method is often called Sobolev smoothing [63, 64, 108]. The smoothed update is still a descent direction, as can be seen through

$$\begin{aligned}\int_{\Sigma} \mathcal{G}\,\boldsymbol{\mathcal{V}}\cdot\boldsymbol{\nu}\,\mathrm{d}\sigma &= -\alpha\int_{\Sigma}\mathcal{G}\,\tilde{\mathcal{G}}\,\mathrm{d}\sigma \\ &= -\alpha\int_{\Sigma}\left(\tilde{\mathcal{G}}-\epsilon\Delta_{\Sigma}\tilde{\mathcal{G}}\right)\,\tilde{\mathcal{G}}\,\mathrm{d}\sigma \\ &= -\alpha\int_{\Sigma}\left(\tilde{\mathcal{G}}^2+\epsilon\left(\nabla_{\Sigma}\tilde{\mathcal{G}}\right)^2\right)\,\mathrm{d}\sigma \\ &< 0.\end{aligned}$$

This is also immediately clear by interpreting $\tilde{\mathcal{G}}$ as the gradient with respect to the inner product

$$\left(\tilde{\mathcal{G}},\boldsymbol{\mathcal{V}}\cdot\boldsymbol{\nu}\right)_S \equiv \int_{\Sigma}\tilde{\mathcal{G}}\,\boldsymbol{\mathcal{V}}\cdot\boldsymbol{\nu}+\epsilon\nabla_{\Sigma}\tilde{\mathcal{G}}\cdot\nabla_{\Sigma}(\boldsymbol{\mathcal{V}}\cdot\boldsymbol{\nu})\,\mathrm{d}\sigma.$$

When the grid must remain field aligned, design updates are chosen along $\mathbf{e}_\theta$. A convenient design velocity takes the form $\boldsymbol{\mathcal{V}} = -\alpha\tilde{\mathcal{G}}\,\mathbf{e}_\theta/(\mathbf{e}_\theta\cdot\boldsymbol{\nu})$, which has a normal component equal to the smoothed gradient $\tilde{\mathcal{G}}$.

The smoothing algorithm not only ensures regularity of the domain, it also acts as a preconditioner for the design equation. In Ref. [64] it is reported that this smoothing allows to take much larger steps in design space. In Refs. [3, 108, 13], it is shown that in many aerodynamic shape optimization problems the Sobolev smoothing presented above resembles the action of the Hessian, meaning that the smoothed gradient is an approximate Newton direction, tending to increase convergence speed of the optimization algorithm significantly. This will be further elaborated in Section 3.5.

2.4.3 Continuous versus Discrete Adjoint Approach

The derivation of the optimality conditions given above follows the continuous adjoint approach, meaning that the derivation is performed completely analytically. This leads to the formulation of an adjoint PDE and an expression for the shape derivative based on the Hadamard-Zolésio structure theorem. Before these results can be used, a discretization step is required. This method is also called Optimize-Before-Discretize (OBD).

Alternatively, in the discrete adjoint or Discretize-Before-Optimize (DBO) approach the cost functional and state equations are discretized first, leading to an optimization problem of the form

$$\min_{\phi_h \in \phi_{h,\mathrm{ad}}, q_h} \quad J_h(\phi_h, q_h) \quad \text{subject to} \quad B_h(\phi_h, q_h) = 0, \tag{2.38}$$

or in reduced form

$$\min_{\phi_h \in \phi_{h,\mathrm{ad}}} \quad \hat{J}_h(\phi_h) \equiv J_h(\phi_h, q_h(\phi_h)), \tag{2.39}$$

where subscripts h refer to a characteristic size of the grid on which the problem is discretized. ϕ_h is a vector of (discretized) design variables which determines the geometry of the system, $\Omega_h = \Omega_h(\phi_h)$. In discretized form, the state equations $B_h(\phi_h, q_h)$ are a set of nonlinear algebraic equations for the discretized state variables q_h.

Deriving optimality conditions for the discrete system is conceptually rather straightforward, see for example Refs. [52, 121]. Defining the derivative in direction ϕ_h' as

$$\hat{J}_h'(\phi_h)\phi_h' = \lim_{t \to 0+} \frac{\hat{J}_h(\phi_h + t\phi_h') - \hat{J}_h(\phi_h)}{t},$$

the derivative of the reduced cost functional follows from the chain rule,

$$\hat{J}_h'\phi_h' = \left(\nabla_{\phi_h} J_h\right)^T \phi_h' + \left(\nabla_{q_h} J_h\right)^T q_h',$$

where q_h' is determined by the linearized state constraint

$$B_{h,q_h} q_h' + B_{h,\phi_h}\phi_h' = 0 \quad \Rightarrow \quad q_h' = -\left(B_{h,q_h}\right)^{-1} B_{h,\phi_h}\phi_h'. \tag{2.40}$$

From this expression, it is seen that q_h' is expensive to evaluate, because it means solving a (linearized) state problem for every possible design perturbation ϕ_h'. However, introducing q_h^* as the solution to the dual or adjoint problem

$$\nabla_{q_h} J_h + \left(B_{h,q_h}\right)^T q_h^* = 0,$$

the gradient of the cost functional becomes

$$\nabla \hat{J}_h = \nabla_{\phi_h} J_h + \left(B_{h,\phi_h}\right)^T q_h^*.$$

At the optimum, the derivative of the cost functional must be non negative for feasible directions ϕ_h', so the discrete optimality conditions are

$$\left\{ \begin{array}{lcll} 0 & = & B_h(\phi_h, q_h) & \text{state equations,} \\ 0 & = & \nabla_{q_h} J_h + \left(B_{h,q_h}\right)^T q_h^* & \text{adjoint equations,} \\ 0 & \leq & \left(\nabla_{\phi_h} J_h + \left(B_{h,\phi_h}\right)^T q_h^*\right)^T \phi_h' & \text{design equation.} \end{array} \right. \tag{2.41}$$

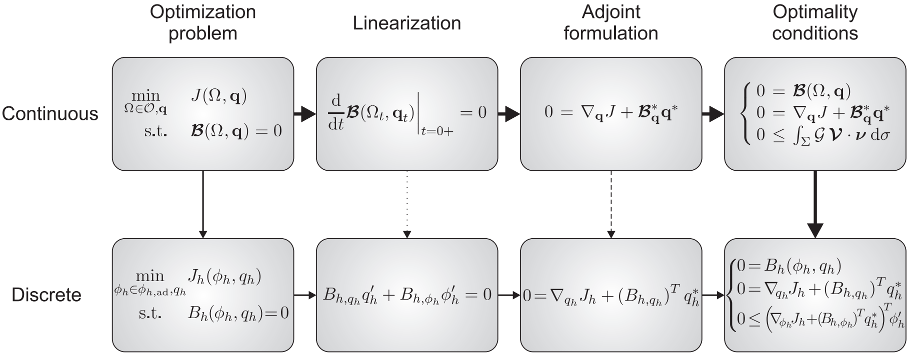


Figure 2.6: Continuous versus discrete adjoint approach. Figure based on Refs. [47, 30].

As in the continuous case, these conditions may also be reinterpreted in terms of the derivatives of the Lagrangian L_h,

$$L_h = J_h(\phi_h, q_h) + (q_h^*, B_h(\phi_h, q_h)) \,. \tag{2.42}$$

There is a direct correspondence between the optimality conditions of the continuous and discrete adjoint approaches. This correspondence is further illustrated in Fig. 2.6. In this figure, the thick solid arrows indicate the continuous adjoint approach through shape sensitivity analysis, while the thin solid arrows represent the discrete adjoint approach. Note that also hybrid approaches, indicated by the dashed and dotted arrows, are possible. In principle, the continuous and discrete adjoint approaches differ only in discretization effects, and should therefore give the same results in the limit of infinite grid resolution. In actual implementations with finite grid size, there may be some important practical consequences. Several authors have compared the performance and accuracy of the two methods, see for example Refs. [47, 84, 85, 43]. Each of the methods has its own advantages and disadvantages, but there is no general consensus as to which of the two is better. In summary, the following items are generally considered advantages of the discrete adjoint method:

- The discrete approach gives the exact derivatives of the discretized optimization problem, which allows convergence to the true solution of the discretized problem. In the continuous adjoint case, there may be a discrepancy between the continuous and discrete derivatives close to the minimum, so derivatives computed with the continuous adjoint method may no longer be a descent direction for the discrete problem. This has to be taken into account for example when performing a line search on the value of the discretized cost functional. Furthermore, the accuracy of the implementation can be tested more easily in the discrete case. In the continuous case, a difference between the adjoint and finite difference sensitivities could be either due to discretization effects or programming errors. The latter can only be excluded by grid sensitivity analysis.

- The discrete adjoint method has identical asymptotic convergence behavior, since the discrete algorithm is the exact transpose of the forward problem, and thus its matrix has the same eigenvalues. In the continuous case, a lot of insight in the problem may be needed to achieve satisfactory convergence speed. Here, the discrete adjoint problem may serve as a guideline for implementing the continuous problem.
- When the underlying PDE models are complex, deriving the continuous adjoint equations and boundary conditions may become difficult. This is especially true for kinetic corrections as flux limiters.

The main benefits of the continuous adjoint method are:

- The continuous adjoint method gives theoretical insight into the nature of the problem. For example, it provides information on the direction in which characteristics in the solution propagate, see Chapter 3. Also, the consistency of the boundary conditions can be verified, since ill posed boundary conditions in the forward problem will not lead to a well defined set of adjoint boundary conditions. Furthermore, the continuous adjoint problem allows to identify hidden assumptions in the modeling which could not be identified by the discrete adjoint method. An example of this will be given in Chapter 4, when studying the effect of neglecting grid non-orthogonality.
- The continuous adjoint program is usually easier in implementation. The programmer is free to choose the discretization schemes. Often, much of the coding from the forward equations can be reused. In the discrete case, all discretization schemes in the forward problem have to be linearized exactly, line by line, which may be tedious and error prone in complex codes as B2-EIRENE. Although in principle this could be done ‘automatically’ by AD software as ADIFOR [10] or ADOL-C [50], in practice some parts of the code (e.g. intermediate matrix solves) may be difficult to handle in this way. An additional difficulty is hidden in the mapping from design variables to the grid: also the entire grid generator has to be differentiated, which may be an enormous task. This has been done for example in Ref. [40]. In practice, one may have to resort to finite differencing to compute the ‘mesh Jacobian’, but this turns out to be very expensive

and may consume a large part of the advantage of using the adjoint method. The velocity method used in this thesis completely circumvents this last issue. Furthermore, since the Hadamard shape derivatives involve boundary data only, avoiding the ‘adjoint grid generator’ makes shape sensitivity computation even cheaper. This is especially useful for the one-shot method discussed in Chapter 3.

- The continuous program may be more robust in the presence of stochastic noise which arises in the coupling of fluid and kinetic (Monte Carlo) systems, see Chapter 8. In the continuous adjoint method, exact derivatives of noisy data are not required explicitly. In particular, it is not needed to differentiate the trajectories of individual Monte Carlo particles. Instead, an adjoint kinetic system is derived analytically. On the other hand, in Chapter 7 it is shown that for well resolved kinetic simulations, the discrete approach can be advantageous.

2.5 Summary

This chapter has presented a general, theoretical formulation of the divertor design problem as a shape optimization problem. A wide body of literature on optimal shape design is available, mainly from the field of computational aerodynamics. A brief overview of the different approaches is given. Based on the specific properties of the edge plasma application, the nonparametric approach through shape sensitivity analysis is selected.

It is shown that efficient computation of shape sensitivities is possible by using the Hadamard-Zolésio structure theorem. The shape sensitivities depend only on boundary data, and do not require the actual perturbation of design variables and remeshing the domain. In order to treat the state equation constraint efficiently, the adjoint equations naturally appear.

In a following step, the first order optimality conditions are presented. These conditions characterize a solution to the design optimization problem. Additional important issues as gradient smoothing are introduced.

Throughout the thesis, the optimization approach will be applied to a variety of representative design problems. In Chapters 3–5, the methodology is used in combination with a fluid edge plasma model, and optimal divertor design is explored both in simplified and in realistic edge plasma geometry. In Chapters 6–8, the methodology will be further extended to coupled fluid-kinetic systems of equations. Applications to a 1D edge plasma model with kinetic neutrals and a 2D (fluid) edge plasma model with additional radiation transport are treated.

Chapter 3

Elements of the Optimization Problem

In this chapter, the essential ingredients of a design code are discussed. The first important component is a model of the system, which will be described in Section 3.1. For initial design studies, it is desirable to have a model which describes the relevant physical processes with sufficient accuracy, yet is fast enough to allow for quick computations. Therefore, the models used in this thesis are based on the models used in edge codes such as B2-EIRENE [96], but make additional simplifying assumptions. As a main simplification compared to full edge codes, the neutral component is not treated kinetically. Instead, the neutral density is obtained through an isotropic pressure diffusion equation. On the one hand, this leads to significant savings in terms of computational time. On the other hand, a fluid neutral model does not suffer from the statistical noise which is introduced by Monte Carlo codes, and is thus a better starting point for the the application of adjoint methods. In Section 3.2, a more general formulation of the model as a set of coupled convection-diffusion equations will be given. This will allow for an elegant derivation of the adjoint equations, and direct applicability of the results to the larger sets of PDEs wich are solved by for example B2.

The second part of the design code is a tool which is able to efficiently and accurately compute design sensitivities. In Chapter 2, it has already been indicated that this calls for an adjoint approach. The adjoint equations will be derived and discussed in Section 3.3, first in the general convection-diffusion framework, and then detailed to the edge plasma model described in Section 3.1.

Section 3.4 covers the details of implementation of the state and adjoint equations in a finite volume code.

Finally, the third element of a design code — the algorithm which uses the model and the (shape) sensitivity information to automatically construct the optimal solution — is discussed. Different candidate algorithms will be described and compared in Section 3.5.

3.1 Edge Plasma Model

3.1.1 Coordinate Systems of the Edge Plasma

When simulating the plasma edge of a tokamak with transport codes, it is customary to assume toroidal symmetry. This symmetry can be exploited most easily in a poloidal-radial-toroidal (θ, r, ϕ) coordinate system, see Fig. 3.1, because all derivatives in the toroidal ϕ direction will be zero. Therefore, the problem becomes two-dimensional and can be described in the poloidal plane. On the other hand, the anisotropy in transport parallel and perpendicular to the magnetic field can be described conveniently in a parallel-diamagnetic-radial $(||, \perp, r)$ coordinate system.

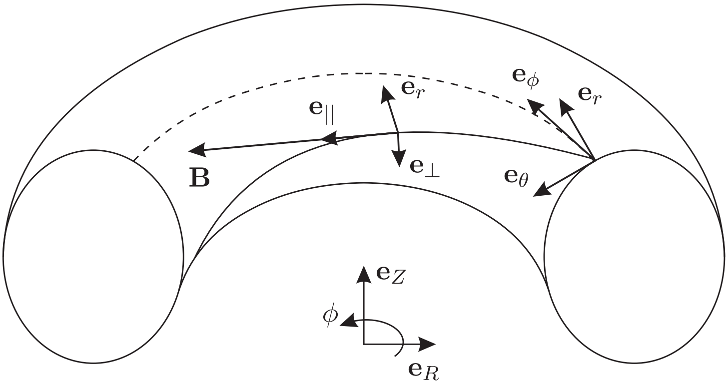


Figure 3.1: Definition of different curvilinear coordinate systems.

Both the (θ, r, ϕ) and $(||, \perp, r)$ systems are right-handed, orthogonal, curvilinear coordinate systems. The parallel $||$ direction is parallel to the magnetic field, and the poloidal θ direction lies in the poloidal plane, parallel to the poloidal projection of the magnetic field. The sine of the angle between the total magnetic field $\mathbf{B}$ and its toroidal component $\mathbf{B}_\phi$ is called the pitch, $b_\theta = \frac{B_\theta}{B}$.

Similarly, $b_\phi = \frac{B_\phi}{B}$, and $b_\theta^2 + b_\phi^2 = 1$. The radial and diamagnetic directions are perpendicular to the magnetic field. The former is also perpendicular to the magnetic flux surface, while the latter is tangent to the flux surface.

A general vector $\mathbf{V} = V_\theta \mathbf{e}_\theta + V_r \mathbf{e}_r + V_\phi \mathbf{e}_\phi = V_{||}\mathbf{e}_{||} + V_\perp \mathbf{e}_\perp + V_r \mathbf{e}_r$ can easily be expressed in both coordinate systems by noting the systems are only rotated around the radial axis with respect to each other. Therefore,

$$V_\theta = b_\theta V_{||} + b_\phi V_\perp, \tag{3.1}$$

$$V_\phi = b_\phi V_{||} - b_\theta V_\perp, \tag{3.2}$$

or the inverse relation

$$V_{||} = b_\theta V_\theta + b_\phi V_\phi, \tag{3.3}$$

$$V_\perp = b_\phi V_\theta - b_\theta V_\phi. \tag{3.4}$$

3.1.2 Plasma Continuity and Parallel Momentum Equations

A single species (Deuterium) plasma is considered, with ions of mass $m \equiv m_\mathrm{i}$ and charge state Z_i. Charge neutrality is assumed, so the ion and electron densities n_i and n_e are related by $n_\mathrm{e} = Z_\mathrm{i} n_\mathrm{i}$. The continuity equation for the plasma can thus be written in terms of the plasma density[1] $n \equiv n_\mathrm{i}$,

$$\frac{\partial n}{\partial t} + \nabla \cdot (n\mathbf{V}_\mathrm{i}) = S_{n_\mathrm{i}}. \tag{3.5}$$

The particle source S_{n_i} due to interactions with the neutrals is

$$S_{n_\mathrm{i}} = n_\mathrm{e} n_\mathrm{n} K_\mathrm{i} - n_\mathrm{i} n_\mathrm{e} K_\mathrm{r}, \tag{3.6}$$

with K_i and K_r the rate coefficients for ionization and volume recombination and n_n the neutral density. For use in the energy equation below, also the ion and electron particle fluxes are defined,

$$\mathbf{\Gamma}^\mathrm{i} = n_\mathrm{i}\mathbf{V}_\mathrm{i},$$

$$\mathbf{\Gamma}^\mathrm{e} = n_\mathrm{e}\mathbf{V}_\mathrm{e},$$

---

[1] In the exposition of the equations, the variable n will be used as main density variable. In cases where the distinction between ion and electron density is important, as in the collision terms, n_i and n_e will be used explicitly.

with $\mathbf{V}_\mathrm{i}$ and $\mathbf{V}_\mathrm{e}$ the ion and electron velocities, respectively. Assuming currents are zero, $\mathbf{V}_\mathrm{i} = \mathbf{V}_\mathrm{e}$ and $Z_\mathrm{i}\mathbf{\Gamma}^\mathrm{i} = \mathbf{\Gamma}^\mathrm{e}$.

Due to the anisotropy in physical processes, it is interesting to decompose the momentum equation in its components for the parallel, diamagnetic and radial velocities, $\mathbf{V}_\mathrm{i} = (u_{||}, u_\perp, u_r)^T$, see Fig. 3.1. The parallel ion (and electron) velocity $u_{||}$ is then described by the total (tensorial) momentum equation projected onto the parallel direction,

$$\frac{\partial}{\partial t}\left(mnu_{||}\right) + \nabla \cdot \left(mnu_{||}\mathbf{V}_\mathrm{i} - \eta^\mathrm{i}\nabla u_{||}\right) = S_{mu_{||}} - \nabla_{||}p, \tag{3.7}$$

where $\nabla_{||} \equiv \mathbf{e}_{||} \cdot \nabla$. This equation is the sum of ion and electron parallel momentum equations. However, since $m_\mathrm{e} << m$, electron inertia is neglected. Furthermore, because $\eta^\mathrm{e} \sim m_\mathrm{e}^{1/2}$ there is also no contribution from electron viscosity. As in B2, several curvature terms are neglected in the parallel momentum equation. For details, see Ref. [5]. $p = p_\mathrm{i} + p_\mathrm{e} = n_\mathrm{i}T_\mathrm{i} + n_\mathrm{e}T_\mathrm{e}$ is the plasma pressure, and $\eta^\mathrm{i} = \mathrm{diag}(\eta^\mathrm{i}_\theta, \eta^\mathrm{i}_r)$ the (anisotropic) ion viscosity tensor. The parallel momentum source term $S_{mu_{||}}$ is due to momentum gains and losses at ionization and recombination events, respectively. Furthermore, ions and neutrals exchange momentum at charge-exchange collisions, determined by the rate coefficient K_cx. Thus, the complete parallel momentum source term is

$$S_{mu_{||}} = mn_\mathrm{e}n_\mathrm{n}K_\mathrm{i}u_{\mathrm{n}||} - mn_\mathrm{i}n_\mathrm{e}K_\mathrm{r}u_{||} - mn_\mathrm{i}n_\mathrm{n}K_\mathrm{cx}(u_{||} - u_{\mathrm{n}||}), \tag{3.8}$$

with $u_{\mathrm{n}||}$ the parallel neutral velocity.

In principle, an equation for the radial velocity u_r would be found by projecting the tensorial momentum equation onto the radial direction [5]. However, transport in the radial direction is dominated by so-called *anomalous* turbulent processes, which are not completely understood theoretically. The (neo)classical momentum equations tend to underestimate the radial transport observed in experiments. Therefore, the radial momentum equation is usually not solved, but replaced by an empirical anomalous diffusion relation

$$nu_r = -D^\mathrm{i}\nabla_r n, \tag{3.9}$$

with D^i an anomalous diffusion coefficient. This analytical relation can be used directly to eliminate the radial ion velocity from the continuity, momentum and energy equations (see below).

The diamagnetic ion velocity is smaller the larger the tokamak device, and determined by drifts [5]. In this thesis, drifts are neglected, so $u_\perp = 0$, and the corresponding component of the momentum equation is not needed.

3.1.3 Fluid Neutral Model

A brief overview of fluid neutral models

Over the past decades, several fluid neutral models have been used in edge modeling. Usually, only one atomic neutral species is considered in these models, with a particle mass m_{n} equal to the mass of the corresponding ions, m. Early fluid neutral models were based on a diffusion equation for the neutral density n_{n}, derived from one-group neutral diffusion theory by considering ionization and charge-exchange processes with ions [127, 101, 71]:

$$\frac{\partial n_{\mathrm{n}}}{\partial t} - \nabla \cdot \left(D^{\mathrm{n}} \nabla n_{\mathrm{n}}\right) = S_{n_{\mathrm{n}}}.$$

$S_{n_{\mathrm{n}}}$ is the neutral particle source due to ionization (in practice, a sink). The (density) diffusion coefficient D^{n} is derived from the transport cross-section, leading to

$$D^{\mathrm{n}} = \frac{c_{\mathrm{n}}^2}{3 n_{\mathrm{i}} \left(K_{\mathrm{i}} + K_{\mathrm{cx}}\right)},$$

with $c_{\mathrm{n}} = (8 T_{\mathrm{n}}/(\pi m_{\mathrm{n}}))^{1/2}$ the neutral thermal speed and T_{n} the neutral temperature. According to Ref. [72], these diffusion models were not able to capture the 'momentum aspect' of detachment, i.e. the drop in ion saturation current, because the plasma-neutral (parallel) momentum coupling could not be described correctly. This coupling is important in detachment, since a lot of plasma parallel momentum can be transferred to the neutrals, after which it is spread out effectively by neutral viscosity [73, 72].

In order to improve the description of momentum exchange, complete Navier-Stokes models were developed for the neutral velocity $\mathbf{V}_{\mathrm{n}}$ [73, 72, 70, 123], or pressure diffusion relations obtained by only retaining pressure-gradient and charge-exchange terms in the momentum equation, see for example Refs. [102, 123]. In the latter approach, an analytical expression for the neutral velocity is obtained which can be substituted into the neutral continuity equation, leading to

$$\frac{\partial n_{\mathrm{n}}}{\partial t} + \nabla \cdot \left(n_{\mathrm{n}} \mathbf{V}_{\mathrm{n}}\right) = S_{n_{\mathrm{n}}}, \quad \text{with} \quad \mathbf{V}_{\mathrm{n}} = \mathbf{V}_{\mathrm{i}} - \frac{1}{m n_{\mathrm{i}} n_{\mathrm{n}} K_{\mathrm{cx}}} \nabla p_{\mathrm{n}}.$$

$p_{\mathrm{n}} = n_{\mathrm{n}} T_{\mathrm{n}}$ is the neutral pressure. This equation implies a pressure diffusion coefficient D_p^{n} given by

$$D_p^{\mathrm{n}} = \frac{1}{m n_{\mathrm{i}} K_{\mathrm{cx}}}.$$

In Ref. [123], it is reported that a pressure diffusion model is much more robust compared to a full Navier-Stokes model, especially in complicated geometries.

Most present edge codes use the parallel component of the neutral momentum equation to determine $u_{\mathrm{n}||}$, and a pressure diffusion relation for the radial and diamagnetic directions in their fluid neutral models, see e.g. Ref. [129, 97] (UEDGE) and Ref. [20] (SOLPS 5.0). This allows to describe in detail the coupling with the ion parallel velocity, while retaining a simplified expression in the other directions. Note, however, that this makes the neutral model anisotropic. The poloidal and radial neutral particle fluxes, $\Gamma_\theta^{\mathrm{n}}$ and Γ_r^{n}, are

$$\begin{aligned}\Gamma_\theta^{\mathrm{n}} &= b_\theta n_{\mathrm{n}} u_{\mathrm{n}||} - D_p^{\mathrm{n}} \nabla_\theta p_{\mathrm{n}},\\ \Gamma_r^{\mathrm{n}} &= -D_p^{\mathrm{n}} \nabla_r p_{\mathrm{n}}.\end{aligned}$$

In Ref. [97] the pressure diffusion coefficient is

$$D_p^{\mathrm{n}} = \frac{1}{m_{\mathrm{n}} \left(n_{\mathrm{i}} K_{\mathrm{i}} + n_{\mathrm{e}} K_{\mathrm{cx}}\right)},$$

while in Ref. [20], the expression

$$D_p^{\mathrm{n}} = C + \sqrt{T_{\mathrm{i}}/m_{\mathrm{n}}} \left(n_{\mathrm{i}} \sigma_{\mathrm{ion}} + n_{\mathrm{e}} \sigma_{\mathrm{cx}}\right)^{-1}$$

is given, with σ_{ion} and σ_{cx} the microscopic cross-sections for ionization and charge exchange. These pressure diffusion coefficients result from keeping pressure-gradient, charge-exchange and ionization terms in the neutral momentum equation. The constant C takes into account hydrogen gas diffusion.

Validity of fluid neutral models

Many of the fluid neutral models described above have been mutually benchmarked [123], benchmarked to Monte Carlo neutral codes [97, 21, 60, 37] and to experiments [21]. Based on such benchmarks, some corrections to fluid neutral models are suggested in Ref. [21]. In particular, it is advised to flux limit the neutral particle and heat flux, and to modify the plasma boundary conditions at core and wall. The first recommendation has not yet been used in this thesis. The adaptations to the boundary conditions will be discussed below.

Fluid neutral models have been used for both attached and detached plasma simulations. Typically, these models are able to reproduce midplane profiles quite well [21, 60]. For divertor conditions, notably at detachment, on the other hand, the models are less reliable and produce qualitatively different density and temperature profiles compared to kinetic models [60, 37]. Due to the steep gradients appearing in detached conditions, the fluid approximation is not valid close to the targets. For example, with fluid models, an increase in neutral density towards the low temperature zones is noted [37]. This effect is necessary in order to have the gradient of neutral pressure drive the neutrals away from the target, but is not present in kinetic models. Furthermore, the 'inversion' of electron temperature profiles at the targets (i.e. lowest T_e near the separatrix strike point) is not reproduced. Delayed onset of detachment with the fluid model is reported in a.o. Refs. [97, 37]. However, the main features of detachment, such as density roll-over, seem to be present, see a.o. Ref. [97]. Also the peaked power deposition profiles agree quite well. For an interesting overview on the abilities of different fluid neutral models to reproduce features of detachment, see Ref. [72].

Fluid neutral model based on a simplified momentum equation

The fluid neutral model used in this thesis is composed of a continuity equation

$$\frac{\partial n_\mathrm{n}}{\partial t} + \nabla \cdot (n_\mathrm{n} \mathbf{V}_\mathrm{n}) = S_{n_\mathrm{n}}, \tag{3.10}$$

and a simplified neutral momentum equation. It is assumed that the Navier-Stokes equation

$$\frac{\partial}{\partial t} (m_\mathrm{n} n_\mathrm{n} \mathbf{V}_\mathrm{n}) + \nabla \cdot (m_\mathrm{n} n_\mathrm{n} \mathbf{V}_\mathrm{n} \mathbf{V}_\mathrm{n}) = \mathbf{S}_{m\mathbf{V}_\mathrm{n}} - \nabla p_\mathrm{n} - \nabla \cdot \Pi^\mathrm{n},$$

with Π^n the neutral stress tensor, can be reduced to a balance between the pressure-gradient force and the momentum exchange terms with the ions:

$$0 = \mathbf{S}_{m\mathbf{V}_\mathrm{n}} - \nabla p_\mathrm{n} \tag{3.11}$$

with

$$\mathbf{S}_{m\mathbf{V}_\mathrm{n}} = m n_\mathrm{i} n_\mathrm{n} K_\mathrm{cx} (\mathbf{V}_\mathrm{i} - \mathbf{V}_\mathrm{n}) - m n_\mathrm{n} n_\mathrm{e} K_\mathrm{i} \mathbf{V}_\mathrm{n} + m n_\mathrm{i} n_\mathrm{e} K_\mathrm{r} \mathbf{V}_\mathrm{i}.$$

Due to conservation of mass, the neutral particle source in Eq. (3.10) has equal magnitude but different sign compared to the ion source, $S_{n_\mathrm{n}} = -S_{n_\mathrm{i}}$. The same is true for the momentum sources. However, for the ions only the momentum source in the parallel direction is used. This type of simplified momentum equation is at the basis of the fluid neutral models described above. Compared to those models, all momentum exchange terms are fully retained, including the volume recombination term. Similar simplifications of the momentum equation are also frequently used in the modeling of ion impurities, for example in the EMC3 code [35].

The neutral momentum equation (3.11) is an algebraic relation from which the neutral particle flux $\mathbf{\Gamma}^{\mathrm{n}}$ can be determined:

$$\mathbf{\Gamma}^{\mathrm{n}} = n_{\mathrm{n}}\mathbf{V}_{\mathrm{n}} = \frac{n_{\mathrm{n}}K_{\mathrm{cx}} + n_{\mathrm{e}}K_{\mathrm{r}}}{K_{\mathrm{cx}} + Z_{\mathrm{i}}K_{\mathrm{i}}}\mathbf{V}_{\mathrm{i}} - D_p^{\mathrm{n}}\nabla p_{\mathrm{n}}, \tag{3.12}$$

with

$$D_p^{\mathrm{n}} = \frac{1}{m\left(n_{\mathrm{i}}K_{\mathrm{cx}} + n_{\mathrm{e}}K_{\mathrm{i}}\right)}. \tag{3.13}$$

This neutral model thus has an isotropic pressure diffusion coefficient. However, the neutral momentum sources tend to be anisotropic since the plasma velocity in radial and diamagnetic directions can be neglected, so $\mathbf{V}_{\mathrm{i}} \approx u_{||}\mathbf{e}_{||}$. With this approximation, and defining a *weighted* neutral density

$$n_{\mathrm{n,eq}} \equiv \frac{n_{\mathrm{n}}K_{\mathrm{cx}} + n_{\mathrm{e}}K_{\mathrm{r}}}{K_{\mathrm{cx}} + Z_{\mathrm{i}}K_{\mathrm{i}}}, \tag{3.14}$$

substituting Eq. (3.12) into the neutral continuity equation (3.10) gives a modified pressure diffusion equation for the neutrals,

$$\frac{\partial n_{\mathrm{n}}}{\partial t} + \nabla \cdot \left(n_{\mathrm{n,eq}}u_{||}\mathbf{e}_{||} - D_p^{\mathrm{n}}\nabla p_{\mathrm{n}}\right) = S_{n_{\mathrm{n}}}. \tag{3.15}$$

3.1.4 Energy Equation

For each of the components a of the plasma (ions, electrons, neutrals), an equation describing the conservation of total energy can be written. This equation is of the form [5]

$$\frac{\partial}{\partial t}\left(\frac{3}{2}n_aT_a + m_an_a\frac{\mathbf{V}_a^2}{2}\right) + \nabla\cdot\left(\left(\frac{5}{2}T_a + m_a\frac{\mathbf{V}_a^2}{2}\right)\mathbf{\Gamma}^a - \kappa^a\nabla T_a + \Pi^a\cdot\mathbf{V}_a\right) = S_{E_{\mathrm{t}},a}. \tag{3.16}$$

T_a is the temperature of species a, Π^a its stress tensor (in turn determined by the viscosity η^a), and $S_{E_{\mathrm{t}},a}$ the source of total energy for this species. The total energy equation describes the conservation of both internal and kinetic energy. An equation for the latter energy component can be obtained by taking the inner product of the momentum equation for species a with $\mathbf{V}_a$, and subtracting the continuity equation of the species multiplied by $m_a\frac{\mathbf{V}_a^2}{2}$. This leads to

$$\frac{\partial}{\partial t}\left(m_an_a\frac{\mathbf{V}_a^2}{2}\right) + \nabla\cdot\left(m_a\mathbf{\Gamma}^a\frac{\mathbf{V}_a^2}{2} + \Pi^a\cdot\mathbf{V}_a\right) = S_{E_{\mathrm{k}},a} - \mathbf{V}_a\cdot\nabla p_a + \Pi^a : (\nabla\mathbf{V}_a)^T, \tag{3.17}$$

where the source of kinetic energy is

$$S_{E_{\mathrm{k}},a} = \mathbf{V}_a\cdot\mathbf{S}_{m_a\mathbf{V}_a} - m_a\frac{\mathbf{V}_a^2}{2}S_{n_a}. \tag{3.18}$$

Subtracting Eq. (3.17) from Eq. (3.16) leads to an internal energy equation of the form

$$\frac{\partial}{\partial t}\left(\frac{3}{2}n_a T_a\right)+\nabla\cdot\left(\frac{5}{2}\mathbf{\Gamma}^a T_a - \kappa^a \nabla T_a\right) = S_{E,a}+\mathbf{V}_a\cdot\nabla p_a - \Pi^a : (\nabla \mathbf{V}_a)^T, \tag{3.19}$$

with

$$S_{E,a} = S_{E_\mathrm{t},a} - S_{E_\mathrm{k},a} \tag{3.20}$$

the source of internal energy. The viscous term on the right hand side of Eq. (3.19) represents irreversible conversion of kinetic to internal energy due to viscous heating (this term is always positive). The term $\mathbf{V}_a \cdot \nabla p_a$ is reversible conversion of kinetic to internal energy.

In contrast to more complete edge models, it is assumed here that ions and electrons have the same temperature. Furthermore, in fluid neutral models it is usually assumed that ions and neutrals are in thermal equilibrium due to frequent charge-exchange collisions. Thus, electrons, ions and neutrals share the same temperature $T \equiv T_\mathrm{e} = T_\mathrm{i} = T_\mathrm{n}$, and a single energy equation is sufficient in a first approximation. This energy equation is obtained by summation of the individual internal energy equations:

$$\begin{aligned}\frac{\partial}{\partial t}\left(\frac{3}{2}\sum_a n_a T\right) + \nabla\cdot\left(\frac{5}{2}\sum_a \mathbf{\Gamma}^a T - \kappa\nabla T\right)&\\ = \quad S_E + \nabla u_{||}\cdot \eta^\mathrm{i}\cdot\nabla u_{||} + u_{||}\nabla_{||}p + \mathbf{V}_\mathrm{n}\cdot\nabla p_\mathrm{n}&\end{aligned} \tag{3.21}$$

All three components of the plasma contribute to the energy balance through convection and conduction of energy. $\kappa = \sum_a \kappa^a = \mathrm{diag}(\kappa_\theta, \kappa_r)$ is the heat conductivity tensor, with contributions from electrons, ions, and neutrals. As might be expected, the contribution from the neutrals to κ is isotropic, while the magnetized electrons and ions introduce anisotropy: $\kappa_\theta = \kappa_\theta^\mathrm{e} + \kappa_\theta^\mathrm{i} + \kappa^\mathrm{n}$ and $\kappa_r = \kappa_r^\mathrm{e} + \kappa_r^\mathrm{i} + \kappa^\mathrm{n}$. In order to simplify notation, the heat flux $\mathbf{Q}$ is introduced:

$$\mathbf{Q} = \frac{5}{2}\sum_a \mathbf{\Gamma}^a T - \kappa\nabla T,$$

with poloidal and radial components

$$\begin{aligned}Q_\theta &= \frac{5}{2}\sum_a \Gamma_\theta^a T - \kappa_\theta \nabla_\theta T,\\ Q_r &= \frac{5}{2}\sum_a \Gamma_r^a T - \kappa_r \nabla_r T.\end{aligned}$$

Several source terms appear on the right hand side of the energy equation (3.21). Since a global energy balance between all components is considered, energy exchange terms between electrons, ions and neutrals individually cancel. All remaining terms represent either a conversion of kinetic energy to internal energy, or external energy sources or sinks. Furthermore, since only the parallel momentum equation (3.7) is solved for ions and electrons, only the parallel kinetic energy contributes to the total plasma energy equation. Consequently, there are only conversion terms of parallel kinetic energy to internal energy in (3.21), see also Ref. [25]. The last two terms on the right hand side represent the reversible conversion of kinetic to internal energy due to ions, electrons, and neutrals (note that $p = p_\mathrm{i} + p_\mathrm{e}$). The second term is viscous heating due to ion viscosity. A corresponding term could be expected from the neutrals, but is not present because neutral viscosity has been neglected in the momentum equation (3.11) and thus also in the energy equations. Finally, the first term on the right hand side of Eq. (3.21) takes into account contributions from plasma-neutral interactions and impurity radiation,

$$S_E = S_{E_\mathrm{c}} - E_\mathrm{i} n_\mathrm{e} n_\mathrm{n} K_\mathrm{i} - c_z n_\mathrm{i} n_\mathrm{e} L_z, \tag{3.22}$$

$$S_{E_\mathrm{c}} = m\left(\left(u_{||} - u_{\mathrm{n}||}\right)^2 + u_{\mathrm{n}r}^2 + u_{\mathrm{n}\perp}^2\right) K, \tag{3.23}$$

with

$$K \equiv \frac{1}{2} K_\mathrm{i} n_\mathrm{e} n_\mathrm{n} + \frac{1}{2} K_\mathrm{r} n_\mathrm{i} n_\mathrm{e} + K_\mathrm{cx} n_\mathrm{i} n_\mathrm{n}. \tag{3.24}$$

E_i is the (electron) energy loss at an ionization event. This includes the ionization potential E_p, and additional radiation from excited states during the ionization process, $E_{\mathrm{i,r}}$ [5, 116]. The energy of recombination is assumed to be radiated, and is thus not an energy source for the plasma-neutral system. The (electron) energy sink due to impurity radiation is modeled through a radiative loss function $L_z(T)$ and a spatially constant impurity fraction c_z [86]. The internal energy source S_{E_c} arises due to collisions. Although the charge-exchange, ionization and recombination processes conserve total energy, there is a conversion of mean kinetic to internal energy associated with them.

3.1.5 Transport Coefficients

Parallel transport coefficients for ions and electrons are classical according to Braginskii [16]:

$$\begin{aligned}
\eta^{\mathrm{i}}_{||} &= 0.96 n_{\mathrm{i}} T_{\mathrm{i}} \tau_{\mathrm{i}}, \\
\kappa^{\mathrm{e}}_{||} &= 3.2 \frac{n_{\mathrm{e}} T_{\mathrm{e}} \tau_{\mathrm{e}}}{m_{\mathrm{e}}}, \\
\kappa^{\mathrm{i}}_{||} &= 3.9 \frac{n_{\mathrm{i}} T_{\mathrm{i}} \tau_{\mathrm{i}}}{m_{\mathrm{i}}},
\end{aligned}$$

with

$$\begin{aligned}
\tau_{\mathrm{i}} &= \frac{3}{4}\sqrt{\frac{m_{\mathrm{i}}}{\pi}} \frac{T_{\mathrm{i}}^{\frac{3}{2}}}{Z_{\mathrm{i}}^4 n_{\mathrm{i}} \ln\Lambda} \left(\frac{4\pi\epsilon_0}{e^2}\right)^2, \\
\tau_{\mathrm{e}} &= \frac{3}{4}\sqrt{\frac{m_{\mathrm{e}}}{2\pi}} \frac{T_{\mathrm{e}}^{\frac{3}{2}}}{Z_{\mathrm{i}}^2 n_{\mathrm{i}} \ln\Lambda} \left(\frac{4\pi\epsilon_0}{e^2}\right)^2
\end{aligned}$$

the collision times for ion-ion and electron-ion collisions, respectively. ϵ_0 is the permittivity of free space, e the unit of charge, and $\ln\Lambda \approx 10..15$ the Coulomb logarithm. As in B2, a value of 12 is used for $\ln\Lambda$. All these coefficients are highly nonlinear — they scale with $T^{\frac{5}{2}}$. The corresponding poloidal transport coefficients are found by projection onto the poloidal plane,

$$\begin{aligned}
\eta^{\mathrm{i}}_{\theta} &= \frac{4}{3} b_{\theta}^2 \eta^{\mathrm{i}}_{||}, \\
\kappa^{\mathrm{e}}_{\theta} &= b_{\theta}^2 \kappa^{\mathrm{e}}_{||}, \\
\kappa^{\mathrm{i}}_{\theta} &= b_{\theta}^2 \kappa^{\mathrm{i}}_{||}.
\end{aligned}$$

The factor $\frac{4}{3}$ in $\eta^{\mathrm{i}}_{\theta}$ is a neoclassical effect, see a.o. Ref. [5]. Often, these transport coefficients are flux limited because they overestimate transport in the presence of strong gradients. These flux limiters are not used in the present model yet.

In the radial direction, coefficients derived from (neo)classical theory significantly underestimate the transport, which is generally accepted to be mainly due to anomalous turbulent processes. Therefore, coefficients of the form

$$\begin{aligned}\eta_r^{\mathrm{i}} &= \nu^{\mathrm{i}} m n,\\ \kappa_r^{\mathrm{e}} &= \chi^{\mathrm{e}} Z_{\mathrm{i}} n,\\ \kappa_r^{\mathrm{i}} &= \chi^{\mathrm{i}} n\end{aligned}$$

are used, with ν^{i}, χ^{e}, and χ^{i} model parameters to be determined experimentally or from turbulence models. Similarly, the radial ion diffusion coefficient D^{i} is a parameter determined by turbulent transport.

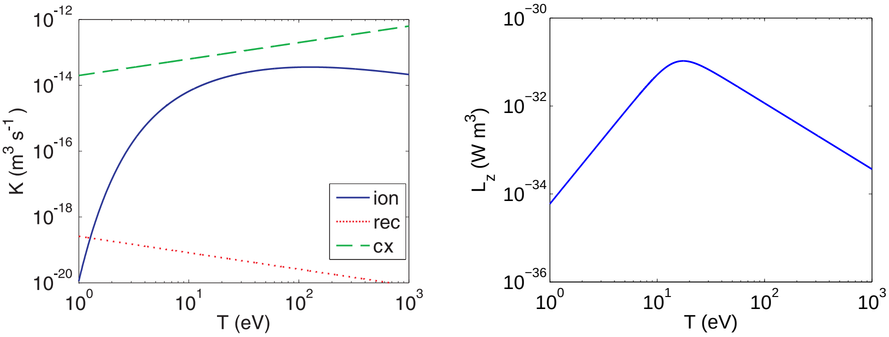
For the neutrals, the pressure diffusion coefficient D_p^{n} follows directly from the momentum equation, see Eq. (3.13). Neutrals also contribute to heat conduction. In accordance with other fluid neutral models, a conduction coefficient of the form $\kappa^{\mathrm{n}} = \chi^{\mathrm{n}} p_{\mathrm{n}} D_p^{\mathrm{n}}$ is used here, with $\chi^{\mathrm{n}} \approx 1.4$. Also for neutral transport coefficients, flux limiters are recommended, see for example Refs. [21, 60]. In Ref. [73], it is advised to correct neutral transport coefficients for neutral-neutral collisions. These effects have not been included yet.

3.1.6 Rate Coefficients and Radiative Loss Function

The source terms in the equations arise from plasma-neutral interactions and impurity radiation. The rate coefficients for electron impact ionization [48], radiative recombination [48], and for charge exchange [105], as well as the radiative loss function L_z of Carbon [86] are approximated using analytical expressions:

$$\begin{aligned}K_{\mathrm{i}} &= \frac{2.0\cdot 10^{-13}}{6.0+\frac{T_{\mathrm{e}}(\mathrm{eV})}{13.6}}\left(\frac{T_{\mathrm{e}}(\mathrm{eV})}{13.6}\right)^{\frac{1}{2}}\exp\left(-\frac{13.6}{T_{\mathrm{e}}(\mathrm{eV})}\right),\\ K_{\mathrm{r}} &= 0.7\cdot 10^{-19}\left(\frac{13.6}{T_{\mathrm{e}}(\mathrm{eV})}\right)^{\frac{1}{2}},\\ K_{\mathrm{cx}} &= 3.2\cdot 10^{-15}\sqrt{\frac{T_{\mathrm{i}}(\mathrm{eV})}{0.026}}, \qquad (3.25)\\ L_z &= \frac{2.0\cdot 10^{-31}}{\left(\frac{T_{\mathrm{e}}(\mathrm{eV})}{15}\right)^{1.5}+\left(\frac{15}{T_{\mathrm{e}}(\mathrm{eV})}\right)^{3}}.\end{aligned}$$

The temperature dependence of these rate coefficients and radiative loss function is shown in Figure 3.2.



(a) Rate coefficients

(b) Radiative loss function

Figure 3.2: Temperature dependence of rate coefficients and radiative loss function.

3.1.7 Boundary Conditions

Targets

At the targets, an electrostatic sheath is formed where the plasma is no longer neutral. The width of this sheath is of the order of the Debye length [116],

$$\lambda_{\mathrm{D}} = \left(\frac{\epsilon_0 T_{\mathrm{e}}}{n_{\mathrm{e}} e^2} \right)^{1/2} .$$

Since the fluid equations for the plasma are not valid inside the sheath, the domain is usually bounded at the sheath entrance rather than at the true target surface — only a small correction since the Debye length is typically of the order of 10^{-5} m. At the sheath entrance, *sheath conditions* are then applied. These require that the parallel ion velocity reaches sound speed, $u_{||} = \pm c_{\mathrm{s}}$, with $c_{\mathrm{s}} = ((T_{\mathrm{i}} + T_{\mathrm{e}})/m)^{1/2} = (2T/m)^{1/2}$ the isothermal plasma sound speed. The sign is chosen in order to have outflow. The treatment of the continuity equation is more complicated, since it is first order in the poloidal (parallel) direction but second order (purely diffusive) in the radial direction. This can be seen by substituting the radial momentum equation (3.9) into the continuity equation (3.5). In principle, if sound speed is assumed at the targets, no information should propagate back into the domain. However, due to the purely diffusive nature of the flow in the radial direction, information *will* propagate back into the domain radially. In edge codes such as B2 this is not an issue, since the non-orthogonality of the grids is neglected. In this thesis, a 9-point stencil is used with correct discretization of the radial flow across inclined boundaries. Therefore, a boundary condition for the radial flow at the targets has to be assumed. The natural extension of the sheath boundary condition, which assumes that ion flow is locally parallel to the magnetic field, is to assume that the radial transport at the sheath entrance is zero. For later convenience, this is formulated as $\mathbf{\Gamma}^{\mathrm{i}} \cdot \boldsymbol{\nu} = \mathbf{\Gamma}_{\mathrm{t}} \cdot \boldsymbol{\nu}$, with $\mathbf{\Gamma}_{\mathrm{t}} \equiv n_{\mathrm{i}} u_\theta \mathbf{e}_\theta$. The ion and electron energy fluxes to the targets are usually determined through sheath transmission coefficients $\delta_{\mathrm{sh}}^{\mathrm{i}}$ and $\delta_{\mathrm{sh}}^{\mathrm{e}}$,

$$Q^{\mathrm{i}} \;=\; \delta_{\mathrm{sh}}^{\mathrm{i}} T_{\mathrm{i}} \mathbf{\Gamma}^{\mathrm{i}} \cdot \boldsymbol{\nu} \qquad \text{and} \qquad Q^{\mathrm{e}} \;=\; \left(\delta_{\mathrm{sh}}^{\mathrm{e}} + \frac{e\phi_{\mathrm{sh}}}{T_{\mathrm{e}}}\right) T_{\mathrm{e}} \mathbf{\Gamma}^{\mathrm{e}} \cdot \boldsymbol{\nu},$$

where ϕ_{sh} is the sheath potential. Typical values are $\delta_{\mathrm{sh}}^{\mathrm{i}} \approx 2.5$, $\delta_{\mathrm{sh}}^{\mathrm{e}} \approx 2$ and $\delta_{\mathrm{sh}}^{\mathrm{pot}} \equiv e\phi_{\mathrm{sh}}/T_{\mathrm{e}} \approx 3.1$ [5, 116]. The total energy flux to the targets is the sum of the ion and electron fluxes. Furthermore, also the influence of the neutrals has to be taken into account. At the target surface, ions and electrons recombine to neutrals and reenter the domain as atoms (or molecules). The fraction of of the ions recycling this way is specified by a recycling coefficient R, so $\mathbf{\Gamma}^{\mathrm{n}} \cdot \boldsymbol{\nu} = -R\mathbf{\Gamma}_{\mathrm{t}} \cdot \boldsymbol{\nu}$. It is assumed that neutrals recycle with a fixed fraction α of the ion energy at the targets. This energy includes the ion energy at the sheath entrance and the sheath potential energy, which is transferred from electrons to ions as they pass through the sheath [116]. Therefore, a total sheath transmission coefficient δ_{sh} can be defined which imposes and energy flux

$$\mathbf{Q} \cdot \boldsymbol{\nu} = \delta_{\mathrm{sh}} T \mathbf{\Gamma}_{\mathrm{t}} \cdot \boldsymbol{\nu} \equiv \mathbf{Q}_{\mathrm{t}} \cdot \boldsymbol{\nu}$$

with

$$\delta_{\mathrm{sh}} = \delta_{\mathrm{sh}}^{\mathrm{i}} + \delta_{\mathrm{sh}}^{\mathrm{e}} + \delta_{\mathrm{sh}}^{\mathrm{pot}} - \alpha R \left(\delta_{\mathrm{sh}}^{\mathrm{i}} + \delta_{\mathrm{sh}}^{\mathrm{pot}}\right).$$

For example, if $R = 1$ and the neutrals retain half the ion energy, $\delta_{\mathrm{sh}} \approx 4.8$. Combined, the boundary conditions applied at the targets are

$$0 = \boldsymbol{\mathcal{C}}_{\mathrm{t}} = \begin{pmatrix} \left(-\mathbf{\Gamma}_{\mathrm{t}} + \mathbf{\Gamma}^{\mathrm{i}}\right) \cdot \boldsymbol{\nu} \\ \pm c_{\mathrm{s}} - u_{||} \\ \left(R\mathbf{\Gamma}_{\mathrm{t}} + \mathbf{\Gamma}^{\mathrm{n}}\right) \cdot \boldsymbol{\nu} \\ \left(-\mathbf{Q}_{\mathrm{t}} + \mathbf{Q}\right) \cdot \boldsymbol{\nu} \end{pmatrix}. \tag{3.26}$$

Core

The core boundary is typically taken as a flux surface sufficiently inward from the separatrix, so that plasma properties can be assumed to be constant on that flux surface. Thus, the plasma becomes essentially one-dimensional. There are several options for boundary conditions. For the ion continuity equation, either the density n_c or the flux can be specified. Similarly, either the temperature or the energy flux $Q_{r,\mathrm{c}}$ can be specified for the energy equation. For all test cases and model problems below, a fixed density is combined with a fixed power from the core, since these are the parameters which are controlled naturally during operation (e.g. by external heating and fueling). By fixing these parameters during optimization, it is also ensured that the operating point of the machine doesn't change. The parallel velocity is assumed to be zero at the core interface. For the neutrals, finding a good boundary condition is perhaps less straightforward. However, for sufficiently large machines, it can be assumed that there will be no neutral flux to the core. These boundary conditions can be written succinctly as

$$0 = \boldsymbol{\mathcal{C}}_\mathrm{c} = \begin{pmatrix} n_\mathrm{c} - n \\ -u_{||} \\ -\Gamma_r^\mathrm{n} \\ Q_{r,\mathrm{c}} - Q_r \end{pmatrix}. \tag{3.27}$$

In Ref. [21], it is recommended to add some neutral leakage towards the core in this boundary condition. This correction is mainly important if neutral penetration in the closed field line region is significant. This will not be the case here, so the correction is not included.

Wall

The outer wall does not coincide with the actual first wall, but with some outermost flux surface. Since this is not a true physical boundary, applying boundary conditions here will always involve some approximation. Typically, decay lengths λ_n and λ_T for plasma density and temperature are specified at this boundary, and the component of the parallel velocity gradient normal to the boundary is set to zero. Because the neutrals are simulated on the same grid as the plasma, a boundary condition for the neutrals has to be specified here as well. Thus, it is assumed that neutrals recycle at this outermost flux surface, and not at the first wall itself. In contrast, Monte Carlo codes such as EIRENE can account for recycling at the first wall, as such codes follow neutral histories on a grid which fully covers the interior of the vessel. The presence of the neutrals is also reflected in the temperature decay length. While plasma density and ion temperature decay lengths are usually roughly the same, this is no longer true if neutrals are included in the total energy balance. Indeed, in the *far SOL*, the neutrals dominate the (radial) energy transport, leading to larger decay lengths than for anomalous radial ion and electron transport only [21]. In summary, the following boundary conditions are applied at the outer wall:

$$0 = \boldsymbol{\mathcal{C}}_{\mathrm{w}} = \begin{pmatrix} -D^{\mathrm{i}}\frac{n}{\lambda_n} - D^{\mathrm{i}}\nabla_r n \\ -\eta_r^{\mathrm{i}}\nabla_r u_{||} \\ R\Gamma_r^{\mathrm{i}} + \Gamma_r^{\mathrm{n}} \\ -\kappa_r\frac{T}{\lambda_T} - \kappa_r\nabla_r T \end{pmatrix}. \tag{3.28}$$

A gas puff can be included rather easily by adding a (constant) particle source S_{p} to the neutral boundary condition: $S_{\mathrm{p}} + R\Gamma_r^{\mathrm{i}} + \Gamma_r^{\mathrm{n}} = 0$. This term will have no influence on the adjoint problem, and is left out in the further derivations.

The need to impose artificial boundary conditions at an arbitrary outermost flux surface can be avoided by extending the (plasma) grid up to the first wall [6, 24]. Recently, this option has become available in B2, but has not been used yet in this thesis.

Private Flux Boundary

The private flux boundary is taken to be coincident with a flux surface close to the dome. It is treated in a similar fashion as the wall boundary. The boundary conditions applied here are

$$0 = \boldsymbol{\mathcal{C}}_{\mathrm{p}} = \begin{pmatrix} -D^{\mathrm{i}}\frac{n}{\lambda_n} + D^{\mathrm{i}}\nabla_r n \\ \eta_r^{\mathrm{i}}\nabla_r u_{||} \\ -\alpha_{\mathrm{p}} c_{\mathrm{n}} n_{\mathrm{n}} - R\Gamma_r^{\mathrm{i}} - \Gamma_r^{\mathrm{n}} \\ -\kappa_r\frac{T}{\lambda_T} + \kappa_r\nabla_r T \end{pmatrix}. \tag{3.29}$$

Since the outwards pointing unit normal is now in the negative r direction, some signs have switched compared to the equivalent expressions at the wall. Additionally, neutral pumping is included here by specifying an absorption coefficient α_{p}. $c_{\mathrm{n}} = (8T/(\pi m))^{1/2}$ is the neutral thermal speed. The absorption coefficient is linked to the volumetric pumping speed L ($\mathrm{m}^3\ \mathrm{s}^{-1}$) by [95]

$$L = 36.38\, A\, \alpha_{\mathrm{p}} \sqrt{\frac{T}{m}},$$

with A the area of the pump entrance in m^2, T in Kelvin and m in a.m.u.

3.2 General Convection-Diffusion Form of the State Equations

In edge plasma codes, the model equations are solved in the poloidal plane, assuming toroidal symmetry. The expansion of the governing equations in the curvilinear, orthogonal, poloidal-radial system is discussed in Section 3.2.1. Next, a general formulation of the state equations as a coupled set of convection-diffusion equations is given, which allows for a succinct and general derivation of the adjoint equations.

3.2.1 State Equations in Poloidal-Radial Coordinate System

In order to formulate the equations in the (θ, r, ϕ) system, derivative operators have to be expanded in terms of this coordinate system. Details on coordinate transformations to general curvilinear systems can be found in for example Ref. [33]. For a full expansion of the B2 equations, see Ref. [5]. Here, the notation with metric coefficients is only briefly introduced for the case of orthogonal coordinate systems.

The relation between an incremental distance $(\mathrm{d}\theta, \mathrm{d}r, \mathrm{d}\phi)^T$ in the (θ, r, ϕ) system and the actual distance $\mathrm{d}\mathbf{s}$ traveled in Euclidean space is determined by the metric coefficients, $\mathrm{d}s_\theta = h_\theta \mathrm{d}\theta$, $\mathrm{d}s_r = h_r \mathrm{d}r$, $\mathrm{d}s_\phi = h_\phi \mathrm{d}\phi$. In a toroidally symmetric system, it is convenient to take $h_\phi = 2\pi R$, with R the local value of the major radius. The transformation between volume elements is given by the determinant of the Jacobian of the transformation, usually denoted by $\sqrt{g}$:

$$\mathrm{d}\boldsymbol{\omega} = \sqrt{g}\,\mathrm{d}\theta\,\mathrm{d}r\,\mathrm{d}\phi = h_\theta h_r h_\phi\,\mathrm{d}\theta\,\mathrm{d}r\,\mathrm{d}\phi = 2\pi R h_\theta h_r\,\mathrm{d}\theta\,\mathrm{d}r\,\mathrm{d}\phi.$$

The surface elements perpendicular to the coordinate directions are given by

$$\mathrm{d}\sigma_\theta = \frac{\sqrt{g}}{h_\theta}\,\mathrm{d}r\,\mathrm{d}\phi, \qquad \mathrm{d}\sigma_r = \frac{\sqrt{g}}{h_r}\,\mathrm{d}\theta\,\mathrm{d}\phi, \quad \text{and} \quad \mathrm{d}\sigma_\phi = \frac{\sqrt{g}}{h_\phi}\,\mathrm{d}\theta\,\mathrm{d}r.$$

The components of the gradient $\nabla\Phi$ of a scalar function $\Phi(\theta, r, \phi)$ are

$$\begin{aligned}
\nabla_\theta \Phi &= \frac{1}{h_\theta}\frac{\partial \Phi}{\partial \theta}, \\
\nabla_r \Phi &= \frac{1}{h_r}\frac{\partial \Phi}{\partial r}, \\
\nabla_\phi \Phi &= \frac{1}{h_\phi}\frac{\partial \Phi}{\partial \phi}.
\end{aligned}$$

In a toroidally symmetric system, the toroidal component is zero. The gradient is again a vector quantity, so also its parallel and diamagnetic components can be found by using equations (3.3)–(3.4):

$$\begin{aligned}\nabla_{||}\Phi &= \frac{b_\theta}{h_\theta}\frac{\partial \Phi}{\partial \theta},\\ \nabla_{\perp}\Phi &= \frac{b_\phi}{h_\theta}\frac{\partial \Phi}{\partial \theta},\end{aligned}$$

where the contributions of $\nabla_\phi \Phi$ have vanished due to toroidal symmetry. For the divergence of a vector $\mathbf{V}$, the relation

$$\nabla \cdot \mathbf{V} = \frac{1}{\sqrt{g}}\frac{\partial}{\partial \theta}\left(\frac{\sqrt{g}}{h_\theta}V_\theta\right) + \frac{1}{\sqrt{g}}\frac{\partial}{\partial r}\left(\frac{\sqrt{g}}{h_r}V_r\right) + \frac{1}{\sqrt{g}}\frac{\partial}{\partial \phi}\left(\frac{\sqrt{g}}{h_\phi}V_\phi\right)$$

is obtained, where again the last term is zero in case of toroidal symmetry.

With these expressions, the model equations can now be translated to the (θ, r, ϕ) coordinate system. Since toroidal symmetry is assumed, all partial derivatives with respect to ϕ are zero. Also, steady-state problems will be considered in this thesis. Therefore, the time derivative terms are left out at this point. Summarizing, equations (3.5), (3.7), (3.15), and (3.21) lead to the ion continuity equation,

$$\frac{1}{\sqrt{g}}\frac{\partial}{\partial \theta}\left(\frac{\sqrt{g}}{h_\theta}nu_\theta\right) + \frac{1}{\sqrt{g}}\frac{\partial}{\partial r}\left(\frac{\sqrt{g}}{h_r}nu_r\right) = S_{n_\mathrm{i}}, \tag{3.30}$$

the ion parallel momentum equation,

$$\begin{aligned}\frac{1}{\sqrt{g}}\frac{\partial}{\partial \theta}\left(\frac{\sqrt{g}}{h_\theta}mnu_\theta u_{||} - \frac{\sqrt{g}}{h_\theta^2}\eta_\theta^\mathrm{i}\frac{\partial u_{||}}{\partial \theta}\right) + &\\ \frac{1}{\sqrt{g}}\frac{\partial}{\partial r}\left(\frac{\sqrt{g}}{h_r}mnu_r u_{||} - \frac{\sqrt{g}}{h_r^2}\eta_r^\mathrm{i}\frac{\partial u_{||}}{\partial r}\right) &= S_{mu_{||}} - \frac{b_\theta}{h_\theta}\frac{\partial p}{\partial \theta},\end{aligned} \tag{3.31}$$

the modified neutral pressure diffusion equation,

$$\frac{1}{\sqrt{g}}\frac{\partial}{\partial \theta}\left(\frac{\sqrt{g}}{h_\theta}n_{\mathrm{n,eq}}u_\theta - \frac{\sqrt{g}}{h_\theta^2}D_p^\mathrm{n}\frac{\partial p_\mathrm{n}}{\partial \theta}\right) - \frac{1}{\sqrt{g}}\frac{\partial}{\partial r}\left(\frac{\sqrt{g}}{h_r^2}D_p^\mathrm{n}\frac{\partial p_\mathrm{n}}{\partial r}\right) = S_{n_\mathrm{n}}, \tag{3.32}$$

and the energy equation,

$$\begin{aligned}&\frac{1}{\sqrt{g}}\frac{\partial}{\partial \theta}\left(\frac{\sqrt{g}}{h_\theta}\frac{5}{2}\sum_a \Gamma_\theta^a T - \frac{\sqrt{g}}{h_\theta^2}\kappa_\theta\frac{\partial T}{\partial \theta}\right) + \frac{1}{\sqrt{g}}\frac{\partial}{\partial r}\left(\frac{\sqrt{g}}{h_r}\frac{5}{2}\sum_a \Gamma_r^a T - \frac{\sqrt{g}}{h_r^2}\kappa_r\frac{\partial T}{\partial r}\right)\\ &= S_E + \eta_\theta^\mathrm{i}\left(\frac{1}{h_\theta}\frac{\partial u_{||}}{\partial \theta}\right)^2 + \eta_r^\mathrm{i}\left(\frac{1}{h_r}\frac{\partial u_{||}}{\partial r}\right)^2 + u_{||}\frac{b_\theta}{h_\theta}\frac{\partial p}{\partial \theta} + u_{\mathrm{n}\theta}\frac{1}{h_\theta}\frac{\partial p_\mathrm{n}}{\partial \theta} + u_{\mathrm{n}r}\frac{1}{h_r}\frac{\partial p_\mathrm{n}}{\partial r}.\end{aligned} \tag{3.33}$$

Using Eq. (3.1) and recalling that the diamagnetic ion velocity is neglected, the poloidal ion velocity is $u_\theta = b_\theta u_{||}$. The simplified radial momentum equation (3.9),

$$\Gamma_r^{\rm i} = n u_r = -D^{\rm i} \frac{1}{h_r} \frac{\partial n}{\partial r}, \tag{3.34}$$

determines the radial convective flow of ions, parallel momentum and ion and electron energy. In order to eliminate u_r as an unknown, Eq. (3.34) will be substituted directly into Eqs. (3.30)–(3.33).

Finally, the components of the neutral fluxes are summarized:

$$\begin{aligned} \Gamma_\theta^{\rm n} &= n_{\rm n} u_{{\rm n}\theta} &= n_{\rm n,eq} u_\theta - D_p^{\rm n} \frac{1}{h_\theta} \frac{\partial p_{\rm n}}{\partial \theta}, \\ \Gamma_r^{\rm n} &= n_{\rm n} u_{{\rm n}r} &= -D_p^{\rm n} \frac{1}{h_r} \frac{\partial p_{\rm n}}{\partial r}. \end{aligned}$$

These components are found by elaboration of Eq. (3.12), using $\mathbf{V}_{\rm i} \approx u_{||}\mathbf{e}_{||}$ and $n_{\rm n,eq}$ from Eq. (3.14). As opposed to the ion diamagnetic velocity, the neutral diamagnetic velocity is not assumed to be zero, but contributes to the poloidal neutral velocity.

3.2.2 State Equations as a Set of Coupled Convection-Diffusion Equations

For an elegant derivation of the adjoint equations, the state equations (3.30)–(3.33) are now written more succinctly as a set of coupled convection-diffusion equations with source terms for the vector of state variables $\mathbf{q} = (n, u_{||}, p_{\rm n}, T)^T$:

$$\begin{aligned} 0 \;=\; \boldsymbol{\mathcal{B}}(\mathbf{q}) \;=\; & S(\mathbf{q}, \nabla_\theta \mathbf{q}, \nabla_r \mathbf{q}) - \frac{1}{\sqrt{g}} \frac{\partial}{\partial \theta} \left(\frac{\sqrt{g}}{h_\theta} C^\theta(\mathbf{q}) - \frac{\sqrt{g}}{h_\theta^2} D^\theta(\mathbf{q}) \frac{\partial \mathbf{q}}{\partial \theta} \right) \\ & - \frac{1}{\sqrt{g}} \frac{\partial}{\partial r} \left(\frac{\sqrt{g}}{h_r} C^r(\mathbf{q}) - \frac{\sqrt{g}}{h_r^2} D^r(\mathbf{q}) \frac{\partial \mathbf{q}}{\partial r} \right). \end{aligned} \tag{3.35}$$

This formulation is very general, and can be extended to an arbitrary number of convection-diffusion equations. The notation $\boldsymbol{\mathcal{B}}(\mathbf{q}) = 0$ in (3.35) refers to the general state equations of Eq. (2.3). Since the neutral pressure diffusion involves the gradient of the neutral pressure $p_{\rm n}$, it is more convenient to work with $p_{\rm n} = n_{\rm n} T$ rather than with $n_{\rm n}$ in the vector of state variables $\mathbf{q}$. However, care is now needed when taking derivatives with respect to T. The vectors $C^\theta(\mathbf{q})$ and $C^r(\mathbf{q})$ represent the poloidal and radial convective fluxes,

$$C^\theta(\mathbf{q}) \quad = \quad \begin{pmatrix} nu_\theta \\ mnu_\theta u_{||} \\ n_{\mathrm{n,eq}} u_\theta \\ \frac{5}{2}((1+Z_{\mathrm{i}})nu_\theta + n_{\mathrm{n,eq}} u_\theta)T \end{pmatrix}, \quad C^r(\mathbf{q}) \quad = \quad \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

In contrast to what may have been expected, $C^r(\mathbf{q}) = 0$. This is due to the description of anomalous radial transport with diffusive relations. $C^r(\mathbf{q})$ is still included in the derivation, because anomalous transport is sometimes modeled by specifying an anomalous convective velocity, which would lead to nonzero contributions in $C^r(\mathbf{q})$. All diffusive terms are summarized by the generalized diffusion matrices,

$$D^\theta(\mathbf{q}) \quad = \quad \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & \eta^{\mathrm{i}}_\theta & 0 & 0 \\ 0 & 0 & D^{\mathrm{n}}_p & 0 \\ 0 & 0 & \frac{5}{2}TD^{\mathrm{n}}_p & \kappa_\theta \end{pmatrix},$$

$$D^r(\mathbf{q}) \quad = \quad \begin{pmatrix} D^{\mathrm{i}} & 0 & 0 & 0 \\ mu_{||}D^{\mathrm{i}} & \eta^{\mathrm{i}}_r & 0 & 0 \\ 0 & 0 & D^{\mathrm{n}}_p & 0 \\ \frac{5}{2}(1+Z_{\mathrm{i}})TD^{\mathrm{i}} & 0 & \frac{5}{2}TD^{\mathrm{n}}_p & \kappa_r \end{pmatrix}.$$

The source terms in the equations are grouped according to their origin as

$$S(\mathbf{q}, \nabla_\theta \mathbf{q}, \nabla_r \mathbf{q}) = S_{\mathrm{n}}(\mathbf{q}, \nabla_\theta \mathbf{q}, \nabla_r \mathbf{q}) + S_z(\mathbf{q}, \nabla_\theta \mathbf{q}, \nabla_r \mathbf{q}) + S_p(\mathbf{q}, \nabla_\theta \mathbf{q}, \nabla_r \mathbf{q}),$$

with

$$S_{\mathrm{n}} \quad = \quad \begin{pmatrix} n_{\mathrm{e}} n_{\mathrm{n}} K_{\mathrm{i}} - n_{\mathrm{i}} n_{\mathrm{e}} K_{\mathrm{r}} \\ mn_{\mathrm{e}} n_{\mathrm{n}} K_{\mathrm{i}} u_{\mathrm{n}||} - mn_{\mathrm{i}} n_{\mathrm{e}} K_{\mathrm{r}} u_{||} - mn_{\mathrm{i}} n_{\mathrm{n}} K_{\mathrm{cx}}(u_{||} - u_{\mathrm{n}||}) \\ n_{\mathrm{i}} n_{\mathrm{e}} K_{\mathrm{r}} - n_{\mathrm{e}} n_{\mathrm{n}} K_{\mathrm{i}} \\ S_{E_{\mathrm{c}}} - E_{\mathrm{i}} n_{\mathrm{e}} n_{\mathrm{n}} K_{\mathrm{i}} \end{pmatrix},$$

$$
\begin{aligned}
S_z &= \begin{pmatrix} 0 \\ 0 \\ 0 \\ -c_z n_\mathrm{i} n_\mathrm{e} L_z \end{pmatrix}, \\
S_p &= \begin{pmatrix} 0 \\ -\frac{b_\theta}{h_\theta}\frac{\partial p}{\partial \theta} \\ 0 \\ \eta_\theta^\mathrm{i}\left(\frac{1}{h_\theta}\frac{\partial u_{||}}{\partial \theta}\right)^2 + \eta_r^\mathrm{i}\left(\frac{1}{h_r}\frac{\partial u_{||}}{\partial r}\right)^2 + \dots \\ \dots + u_{||}\frac{b_\theta}{h_\theta}\frac{\partial p}{\partial \theta} + u_{\mathrm{n}\theta}\frac{1}{h_\theta}\frac{\partial p_\mathrm{n}}{\partial \theta} + u_{\mathrm{n}r}\frac{1}{h_r}\frac{\partial p_\mathrm{n}}{\partial r} \end{pmatrix}.
\end{aligned}
\tag{3.36}
$$

The source S_n arises due to interactions with neutrals, S_z due to interactions with impurities. S_p groups all other terms in the equations.

3.3 Derivation of the Adjoint Equations

In this section, the adjoint equations for the edge plasma model considered in this thesis are derived. The general form of the state equations (3.35) will be used, so the result will be directly applicable not only to the model itself, but also to simplified (or, possibly, more extended) versions of it. Looking back at Chapter 2, Eq. (2.11), the adjoint equations are given by the second of the saddle point equations,

$$
\nabla_\mathbf{q} L(\Omega, \mathbf{q}, \mathbf{q}^*) = \nabla_\mathbf{q} J(\Omega, \mathbf{q}) + \boldsymbol{\mathcal{B}}_\mathbf{q}^*(\Omega, \mathbf{q})\mathbf{q}^* = 0. \tag{3.37}
$$

Thus, the primary aim is to find the operator $\boldsymbol{\mathcal{B}}_\mathbf{q}^*(\Omega, \mathbf{q})$ such that

$$
\left(\boldsymbol{\mathcal{B}}_\mathbf{q}^*(\Omega, \mathbf{q})\mathbf{q}^*, \delta\mathbf{q}\right) = \left(\mathbf{q}^*, \boldsymbol{\mathcal{B}}_\mathbf{q}(\Omega, \mathbf{q})\delta\mathbf{q}\right).
$$

In the elaboration of (3.37), a generic cost functional based on Eqs. (2.1) and (2.2) is considered, including both domain and boundary contributions:

$$
\begin{aligned}
J(\Omega, \mathbf{q}) &= J_1(\Omega, \mathbf{q}) + J_2(\Omega, \mathbf{q}) \\
&= \int_\Omega f(\mathbf{q})\, \mathrm{d}\boldsymbol{\omega} + \int_\Sigma g(\mathbf{q}, \boldsymbol{\nu})\, \mathrm{d}\sigma.
\end{aligned}
\tag{3.38}
$$

3.3.1 Linearized State Equations

By definition, the adjoint operator $\boldsymbol{\mathcal{B}}_\mathbf{q}^*(\Omega, \mathbf{q})$ is the dual of the state equation operator linearized with respect to the state variables, $\boldsymbol{\mathcal{B}}_\mathbf{q}(\Omega, \mathbf{q})$, so the first step is to find an expression for $\boldsymbol{\mathcal{B}}_{\mathbf{q}}(\Omega,\mathbf{q})\delta\mathbf{q}$. For the state equations $\boldsymbol{\mathcal{B}}(\mathbf{q})$ from Eq. (3.35), linearization with respect to $\mathbf{q}$ leads to

$$\begin{aligned}\boldsymbol{\mathcal{B}}_{\mathbf{q}}\delta\mathbf{q} \quad &= \quad S_{\mathbf{q}}\delta\mathbf{q} + S_{\nabla_\theta\mathbf{q}}\frac{1}{h_\theta}\frac{\partial\delta\mathbf{q}}{\partial\theta} + S_{\nabla_r\mathbf{q}}\frac{1}{h_r}\frac{\partial\delta\mathbf{q}}{\partial r} \\ &\quad -\frac{1}{\sqrt{g}}\frac{\partial}{\partial\theta}\left(\frac{\sqrt{g}}{h_\theta}C^\theta_{\mathbf{q}}\delta\mathbf{q} - \frac{\sqrt{g}}{h_\theta^2}\left(D^\theta\frac{\partial\delta\mathbf{q}}{\partial\theta} + D^\theta_{\mathbf{q}}\delta\mathbf{q}\frac{\partial\mathbf{q}}{\partial\theta}\right)\right) \\ &\quad -\frac{1}{\sqrt{g}}\frac{\partial}{\partial r}\left(\frac{\sqrt{g}}{h_r}C^r_{\mathbf{q}}\delta\mathbf{q} - \frac{\sqrt{g}}{h_r^2}\left(D^r\frac{\partial\delta\mathbf{q}}{\partial r} + D^r_{\mathbf{q}}\delta\mathbf{q}\frac{\partial\mathbf{q}}{\partial r}\right)\right). \end{aligned} \tag{3.39}$$

In this notation, a subscript of $\nabla_\theta\mathbf{q}$ or $\nabla_r\mathbf{q}$ means differentiation with respect to the occurrence of a (poloidal or radial) gradient of $\mathbf{q}$. For ease of notation, arguments of $\mathbf{q}$, $\nabla_\theta\mathbf{q}$, and $\nabla_r\mathbf{q}$ are not repeated explicitly.

Similarly, all boundary conditions have to be linearized. In order to distinguish between Dirichlet, Neumann and mixed type boundary conditions, the boundary conditions are written as

$$\boldsymbol{\mathcal{C}}(\mathbf{q},\nabla_\theta\mathbf{q},\nabla_r\mathbf{q}) = 0, \tag{3.40}$$

and their linearization leads to

$$\boldsymbol{\mathcal{C}}_{\mathbf{q}}\delta\mathbf{q} + \boldsymbol{\mathcal{C}}_{\nabla_\theta\mathbf{q}}\frac{1}{h_\theta}\frac{\partial\delta\mathbf{q}}{\partial\theta} + \boldsymbol{\mathcal{C}}_{\nabla_r\mathbf{q}}\frac{1}{h_r}\frac{\partial\delta\mathbf{q}}{\partial r} = 0. \tag{3.41}$$

Further, also the linearized cost functional is needed:

$$\begin{aligned} J_{\mathbf{q}}\delta\mathbf{q} \quad &= \quad (\nabla_{\mathbf{q}}J,\delta\mathbf{q}) \quad = \quad (\nabla_{\mathbf{q}}J_1,\delta\mathbf{q})_\Omega + (\nabla_{\mathbf{q}}J_2,\delta\mathbf{q})_\Sigma \\ &= \quad \int_\Omega f_{\mathbf{q}}\delta\mathbf{q}\,\mathrm{d}\boldsymbol{\omega} + \int_\Sigma g_{\mathbf{q}}\delta\mathbf{q}\,\mathrm{d}\sigma. \end{aligned} \tag{3.42}$$

Combining all elements leads to the linearized Lagrangian,

$$\begin{aligned} L_{\mathbf{q}}\delta\mathbf{q} \quad &= \quad J_{\mathbf{q}}\delta\mathbf{q} + (\mathbf{q}^*,\boldsymbol{\mathcal{B}}_{\mathbf{q}}\delta\mathbf{q})_\Omega + (\mathbf{p}^*,\boldsymbol{\mathcal{C}}_{\mathbf{q}}\delta\mathbf{q})_\Sigma \\ &= \quad \int_\Omega \left(f_{\mathbf{q}} + \mathbf{q}^*\cdot\boldsymbol{\mathcal{B}}_{\mathbf{q}}\right)\delta\mathbf{q}\,\mathrm{d}\boldsymbol{\omega} + \int_\Sigma \left(g_{\mathbf{q}} + \mathbf{p}^*\cdot\boldsymbol{\mathcal{C}}_{\mathbf{q}}\right)\delta\mathbf{q}\,\mathrm{d}\sigma. \end{aligned} \tag{3.43}$$

For each of the state equations, there is a corresponding adjoint variable: $\mathbf{q}^* = (n^*, u^*_{||}, p^*_{\mathrm{n}}, T^*)^T$. Similarly, each boundary condition requires a corresponding adjoint multiplier, $\mathbf{p}^* = (n^*_S, u^*_{||,S}, p^*_{\mathrm{n},S}, T^*_S)^T$.

3.3.2 Adjoint Equations

Now, using the definition (2.7), the adjoint equations can be found by moving all differential operators in (3.43) from $\delta\mathbf{q}$ to $\mathbf{q}^*$ using integration by parts. This is the so-called formal Lagrangian approach [121, 13]. The method is *formal* in the sense that all functions appearing are assumed to exist and be well defined in appropriate function spaces. However, this will not be checked explicitly.

First, the volume integrals of the linearized state equations are treated, leading to

$$\begin{aligned}
(\mathbf{q}^*, \boldsymbol{\mathcal{B}}_{\mathbf{q}}\delta\mathbf{q})_\Omega &= \int_\Omega \mathbf{q}^* \cdot \boldsymbol{\mathcal{B}}_{\mathbf{q}}\delta\mathbf{q}\,\mathrm{d}\boldsymbol{\omega} \\
&= \int_\Omega \delta\mathbf{q} \cdot \boldsymbol{\mathcal{B}}^*_{\mathbf{q}}\mathbf{q}^*\,\mathrm{d}\boldsymbol{\omega} + \int_\Sigma \mathrm{BT}\,\mathrm{d}\sigma \\
&= \left(\boldsymbol{\mathcal{B}}^*_{\mathbf{q}}\mathbf{q}^*, \delta\mathbf{q}\right)_\Omega + \int_\Sigma \mathrm{BT}\,\mathrm{d}\sigma,
\end{aligned} \tag{3.44}$$

with

$$\begin{aligned}
\boldsymbol{\mathcal{B}}^*_{\mathbf{q}}\mathbf{q}^* = {} & S^T_{\mathbf{q}}\mathbf{q}^* - \frac{1}{\sqrt{g}}\frac{\partial}{\partial\theta}\left(\frac{\sqrt{g}}{h_\theta}S^T_{\nabla_\theta\mathbf{q}}\mathbf{q}^*\right) - \frac{1}{\sqrt{g}}\frac{\partial}{\partial r}\left(\frac{\sqrt{g}}{h_r}S^T_{\nabla_r\mathbf{q}}\mathbf{q}^*\right) \\
& + \left(C^\theta_{\mathbf{q}}\right)^T\frac{1}{h_\theta}\frac{\partial\mathbf{q}^*}{\partial\theta} + \frac{1}{\sqrt{g}}\frac{\partial}{\partial\theta}\left(\frac{\sqrt{g}}{h_\theta^2}(D^\theta)^T\frac{\partial\mathbf{q}^*}{\partial\theta}\right) \\
& + \left(C^r_{\mathbf{q}}\right)^T\frac{1}{h_r}\frac{\partial\mathbf{q}^*}{\partial r} + \frac{1}{\sqrt{g}}\frac{\partial}{\partial r}\left(\frac{\sqrt{g}}{h_r^2}(D^r)^T\frac{\partial\mathbf{q}^*}{\partial r}\right) \\
& - \frac{1}{h_\theta}\frac{\partial(\mathbf{q}^*)^T}{\partial\theta}D^\theta_{\mathbf{q}}\frac{1}{h_\theta}\frac{\partial\mathbf{q}}{\partial\theta} - \frac{1}{h_r}\frac{\partial(\mathbf{q}^*)^T}{\partial r}D^r_{\mathbf{q}}\frac{1}{h_r}\frac{\partial\mathbf{q}}{\partial r},
\end{aligned} \tag{3.45}$$

and

$$\begin{aligned}
\mathrm{BT} &= \mathbf{q}^* \cdot \left(S_{\nabla_\theta \mathbf{q}} \delta\mathbf{q}\right) \nu_\theta + \mathbf{q}^* \cdot \left(S_{\nabla_r \mathbf{q}} \delta\mathbf{q}\right) \nu_r \\
&\quad - \mathbf{q}^* \cdot \left(C^\theta_{\mathbf{q}} \delta\mathbf{q} - D^\theta \frac{1}{h_\theta} \frac{\partial \delta\mathbf{q}}{\partial \theta} - D^\theta_{\mathbf{q}} \delta\mathbf{q} \frac{1}{h_\theta} \frac{\partial \mathbf{q}}{\partial \theta} \right) \nu_\theta \\
&\quad - \mathbf{q}^* \cdot \left(C^r_{\mathbf{q}} \delta\mathbf{q} - D^r \frac{1}{h_r} \frac{\partial \delta\mathbf{q}}{\partial r} - D^r_{\mathbf{q}} \delta\mathbf{q} \frac{1}{h_r} \frac{\partial \mathbf{q}}{\partial r} \right) \nu_r \\
&\quad - \delta\mathbf{q} \cdot \left((D^\theta)^T \frac{1}{h_\theta} \frac{\partial \mathbf{q}^*}{\partial \theta} \nu_\theta + (D^r)^T \frac{1}{h_r} \frac{\partial \mathbf{q}^*}{\partial r} \nu_r \right) \qquad (3.46)
\end{aligned}$$

all boundary terms appearing during integration by parts. It has been assumed that $f(\mathbf{q})$ does not involve spatial derivatives of the state variables. In practice, this is possible, and can be taken into account by a treatment similar to the source terms $S(\mathbf{q}, \nabla_\theta \mathbf{q}, \nabla_r \mathbf{q})$.

Now, the linearized Lagrangian (3.43) takes the form

$$\begin{aligned}
L_{\mathbf{q}} \delta\mathbf{q} &= \int_\Omega \left(\nabla_{\mathbf{q}} f + \boldsymbol{\mathcal{B}}^*_{\mathbf{q}} \mathbf{q}^* \right) \cdot \delta\mathbf{q} \, \mathrm{d}\boldsymbol{\omega} + \int_\Sigma \left(g_{\mathbf{q}} + \mathbf{p}^* \cdot \boldsymbol{\mathcal{C}}_{\mathbf{q}} \right) \delta\mathbf{q} + \mathrm{BT} \, \mathrm{d}\sigma \\
&= \left(\nabla_{\mathbf{q}} J_1 + \boldsymbol{\mathcal{B}}^*_{\mathbf{q}} \mathbf{q}^*, \delta\mathbf{q} \right)_\Omega + \int_\Sigma \left(g_{\mathbf{q}} + \mathbf{p}^* \cdot \boldsymbol{\mathcal{C}}_{\mathbf{q}} \right) \delta\mathbf{q} + \mathrm{BT} \, \mathrm{d}\sigma.
\end{aligned}$$

The integral across the domain already has the desired form. Next, attention is paid to the boundary integral. There are three types of terms contributing to this integral, stemming from 1) a possible cost functional on the boundary, 2) the boundary conditions of the forward problem, and 3) integration by parts.

Grouping terms in $\delta\mathbf{q}$, $\frac{1}{h_\theta}\frac{\partial\delta\mathbf{q}}{\partial\theta}$ and $\frac{1}{h_r}\frac{\partial\delta\mathbf{q}}{\partial r}$ leads to

$$\begin{aligned}
&g_{\mathbf{q}}\delta\mathbf{q} + \mathbf{p}^* \cdot \left(\boldsymbol{\mathcal{C}}_{\mathbf{q}}\delta\mathbf{q} + \boldsymbol{\mathcal{C}}_{\nabla_\theta\mathbf{q}}\frac{1}{h_\theta}\frac{\partial\delta\mathbf{q}}{\partial\theta} + \boldsymbol{\mathcal{C}}_{\nabla_r\mathbf{q}}\frac{1}{h_r}\frac{\partial\delta\mathbf{q}}{\partial r}\right) + \mathrm{BT} \\
&= \left(\nabla_{\mathbf{q}} g + \boldsymbol{\mathcal{C}}_{\mathbf{q}}^T\mathbf{p}^* + \nu_\theta S^T_{\nabla_\theta\mathbf{q}}\mathbf{q}^* + \nu_r S^T_{\nabla_r\mathbf{q}}\mathbf{q}^*\right)\cdot\delta\mathbf{q} \\
&\quad - \nu_\theta\left((C^\theta_{\mathbf{q}})^T\mathbf{q}^* - \mathbf{q}^*\cdot D^\theta_{\mathbf{q}}\frac{1}{h_\theta}\frac{\partial\mathbf{q}}{\partial\theta} + (D^\theta)^T\frac{1}{h_\theta}\frac{\partial\mathbf{q}^*}{\partial\theta}\right)\cdot\delta\mathbf{q} \\
&\quad - \nu_r\left((C^r_{\mathbf{q}})^T\mathbf{q}^* - \mathbf{q}^*\cdot D^r_{\mathbf{q}}\frac{1}{h_r}\frac{\partial\mathbf{q}}{\partial r} + (D^r)^T\frac{1}{h_r}\frac{\partial\mathbf{q}^*}{\partial r}\right)\cdot\delta\mathbf{q} \\
&\quad + \left(\boldsymbol{\mathcal{C}}^T_{\nabla_\theta\mathbf{q}}\mathbf{p}^* + \nu_\theta(D^\theta)^T\mathbf{q}^*\right)\cdot\frac{1}{h_\theta}\frac{\partial\delta\mathbf{q}}{\partial\theta} \\
&\quad + \left(\boldsymbol{\mathcal{C}}^T_{\nabla_r\mathbf{q}}\mathbf{p}^* + \nu_r(D^r)^T\mathbf{q}^*\right)\cdot\frac{1}{h_r}\frac{\partial\delta\mathbf{q}}{\partial r}.
\end{aligned} \tag{3.47}$$

It is noted again that g is assumed to depend only on $\mathbf{q}$ itself, not on spatial derivatives of $\mathbf{q}$. This extension could be done rather easily. Applying integration by parts again to the last two terms of expression (3.47) will lead to the form

$$\left(\nabla_{\mathbf{q}} J_2 + \boldsymbol{\mathcal{C}}^*_{\mathbf{q}}(\Omega,\mathbf{q})\mathbf{q}^*, \delta\mathbf{q}\right)_\Sigma, \tag{3.48}$$

so the linearized Lagrangian is finally

$$L_{\mathbf{q}}\delta\mathbf{q} = \left(\nabla_{\mathbf{q}} J_1 + \boldsymbol{\mathcal{B}}^*_{\mathbf{q}}\mathbf{q}^*, \delta\mathbf{q}\right)_\Omega + \left(\nabla_{\mathbf{q}} J_2 + \boldsymbol{\mathcal{C}}^*_{\mathbf{q}}\mathbf{q}^*, \delta\mathbf{q}\right)_\Sigma.$$

‘Conservative’ form of the adjoint equations

In contrast to the state equations, the convective terms in the adjoint operator $\boldsymbol{\mathcal{B}}^*_{\mathbf{q}}\mathbf{q}^*$ from (3.45) are not in conservative form. This form can be mimicked by rewriting the equation as

$$\begin{aligned}
\boldsymbol{\mathcal{B}}^*_{\mathbf{q}}\mathbf{q}^* &= S^T_{\mathbf{q}}\mathbf{q}^* - \frac{1}{\sqrt{g}}\frac{\partial}{\partial\theta}\left(\frac{\sqrt{g}}{h_\theta}S^T_{\nabla_\theta \mathbf{q}}\mathbf{q}^*\right) - \frac{1}{\sqrt{g}}\frac{\partial}{\partial r}\left(\frac{\sqrt{g}}{h_r}S^T_{\nabla_r \mathbf{q}}\mathbf{q}^*\right) \\
&+\frac{1}{\sqrt{g}}\frac{\partial}{\partial\theta}\left(\frac{\sqrt{g}}{h_\theta}\left(C^\theta_{\mathbf{q}}\right)^T\mathbf{q}^* + \frac{\sqrt{g}}{h_\theta^2}(D^\theta)^T\frac{\partial \mathbf{q}^*}{\partial\theta}\right) \\
&+\frac{1}{\sqrt{g}}\frac{\partial}{\partial r}\left(\frac{\sqrt{g}}{h_r}\left(C^r_{\mathbf{q}}\right)^T\mathbf{q}^* + \frac{\sqrt{g}}{h_r^2}(D^r)^T\frac{\partial \mathbf{q}^*}{\partial r}\right) \\
&-\frac{1}{\sqrt{g}}\frac{\partial}{\partial\theta}\left(\frac{\sqrt{g}}{h_\theta}\left(C^\theta_{\mathbf{q}}\right)^T\right)\mathbf{q}^* - \frac{1}{\sqrt{g}}\frac{\partial}{\partial r}\left(\frac{\sqrt{g}}{h_r}\left(C^r_{\mathbf{q}}\right)^T\right)\mathbf{q}^* \\
&-\frac{1}{h_\theta}\frac{\partial(\mathbf{q}^*)^T}{\partial\theta}D^\theta_{\mathbf{q}}\frac{1}{h_\theta}\frac{\partial\mathbf{q}}{\partial\theta} - \frac{1}{h_r}\frac{\partial(\mathbf{q}^*)^T}{\partial r}D^r_{\mathbf{q}}\frac{1}{h_r}\frac{\partial\mathbf{q}}{\partial r}. \qquad (3.49)
\end{aligned}$$

Of course, this does not make the equations conservative, because additional source terms appear in the equations. The advantage of using this form of the equations is that computational routines which have been written originally for the state equations, can be reused directly for the adjoint equations. In this way, code duplication and potential sources of bugs are eliminated. The additional source terms are easily accommodated.

Discussion

According to (3.37), the adjoint equations are defined by

$$\nabla_{\mathbf{q}}L(\Omega,\mathbf{q},\mathbf{q}^*) = \nabla_{\mathbf{q}}J(\Omega,\mathbf{q}) + \boldsymbol{\mathcal{B}}^*_{\mathbf{q}}(\Omega,\mathbf{q})\mathbf{q}^* = 0.$$

In the previous section, this equation has been elaborated as

$$\begin{aligned}
L_{\mathbf{q}}\delta\mathbf{q} &= (\nabla_{\mathbf{q}}L, \delta\mathbf{q}) \\
&= \left(\nabla_{\mathbf{q}}J_1 + \boldsymbol{\mathcal{B}}^*_{\mathbf{q}}\mathbf{q}^*, \delta\mathbf{q}\right)_\Omega + \left(\nabla_{\mathbf{q}}J_2 + \boldsymbol{\mathcal{C}}^*_{\mathbf{q}}\mathbf{q}^*, \delta\mathbf{q}\right)_\Sigma .
\end{aligned}$$

Thus, the adjoint field equations can be identified with

$$\nabla_{\mathbf{q}}J_1(\Omega,\mathbf{q}) + \boldsymbol{\mathcal{B}}^*_{\mathbf{q}}(\mathbf{q})\mathbf{q}^* = 0 \quad \text{in} \quad \Omega, \qquad (3.50)$$

while the corresponding boundary conditions follow from

$$\nabla_{\mathbf{q}}J_2(\Omega,\mathbf{q}) + \boldsymbol{\mathcal{C}}^*_{\mathbf{q}}(\Omega,\mathbf{q})\mathbf{q}^* = 0 \quad \text{on} \quad \Sigma. \qquad (3.51)$$

A cost functional of type J_1 — which is an integral across the domain depending on $\mathbf{q}$ — leads to a forcing term in the adjoint field equations, while cost functionals of type J_2 — defined on the boundary — lead to forcing terms in the adjoint boundary conditions.

A closer inspection of the adjoint equations (3.49) reveals that convective terms have switched sign compared to the forward equations. Thus, the characteristics in the solution have reversed, which can be interpreted as the adjoint field tracing information backwards towards its origin. Furthermore, the adjoint equations take all linearized effects into account, including the dependence of transport coefficients on the state of the plasma. This leads to additional source terms in the adjoint field equations, see (3.49).

The adjoint boundary conditions are composed of different parts. Indeed, the relation (3.51) above will hold if the three factors multiplying $\delta\mathbf{q}$, $\frac{1}{h_\theta}\frac{\partial\delta\mathbf{q}}{\partial\theta}$ and $\frac{1}{h_r}\frac{\partial\delta\mathbf{q}}{\partial r}$ in (3.47) are zero simultaneously:

$$\left\{\begin{array}{rcl} 0 & = & \nabla_{\mathbf{q}} g + \boldsymbol{\mathcal{C}}_{\mathbf{q}}^T \mathbf{p}^* + \nu_\theta S^T_{\nabla_\theta \mathbf{q}} \mathbf{q}^* + \nu_r S^T_{\nabla_r \mathbf{q}} \mathbf{q}^* \\ & & -\nu_\theta \left((C^\theta_{\mathbf{q}})^T \mathbf{q}^* - \mathbf{q}^* \cdot D^\theta_{\mathbf{q}} \frac{1}{h_\theta} \frac{\partial \mathbf{q}}{\partial \theta} + (D^\theta)^T \frac{1}{h_\theta} \frac{\partial \mathbf{q}^*}{\partial \theta} \right) \\ & & -\nu_r \left((C^r_{\mathbf{q}})^T \mathbf{q}^* - \mathbf{q}^* \cdot D^r_{\mathbf{q}} \frac{1}{h_r} \frac{\partial \mathbf{q}}{\partial r} + (D^r)^T \frac{1}{h_r} \frac{\partial \mathbf{q}^*}{\partial r} \right), \\ & & \\ 0 & = & \boldsymbol{\mathcal{C}}^T_{\nabla_\theta \mathbf{q}} \mathbf{p}^* + \nu_\theta (D^\theta)^T \mathbf{q}^*, \\ & & \\ 0 & = & \boldsymbol{\mathcal{C}}^T_{\nabla_r \mathbf{q}} \mathbf{p}^* + \nu_r (D^r)^T \mathbf{q}^*. \end{array}\right. \tag{3.52}$$

Apart from the adjoint boundary conditions, this also gives conditions relating the Lagrange multipliers for the boundary conditions $\mathbf{p}^*$ to the adjoint variables $\mathbf{q}^*$. The multipliers $\mathbf{p}^*$ are needed for the third optimality condition (see Eq. (2.36)), i.e. in the derivative of the Lagrangian with respect to the control variables. Looking at the three sets of conditions (3.52), it appears that there are more equations than unknowns (3 times 4 equations, for only 2 times 4 unknowns). This is because no direct restrictions have been applied to possible boundary conditions in terms of specification of fluxes (or corresponding gradients). For a well posed problem, only the normal flux at a boundary can be specified. Also in the terms coming from BT, only normal fluxes and gradients appear. Therefore, for boundaries which are perpendicular to one of the coordinate directions, one of the last two sets of equations in (3.52) will become a trivial identity. For boundaries which are not perpendicular to either of the coordinate directions, the last two sets of equations will turn out to be equivalent.

3.3.3 Adjoint Equations in Poloidal-Radial Coordinate System

The tools are now at hand to elaborate the adjoint equations for the edge plasma model described in Section 3.1. The resulting expressions are quite lengthy, but are found directly by elaboration of (3.50) using the matrices defined in (3.35). For ease of implementation later on, the conservative form (3.49) of the adjoint equations is used. Indeed, in this way several standard convection-diffusion type terms appear in the adjoint equations, for which computational routines developed for the forward problem can be reused. Analogously, the boundary conditions to the adjoint equations are found from Eqs. (3.52). Again, the matrices defined by (3.35) are needed, as well as the boundary conditions of the forward problem discussed in Section 3.1.7.

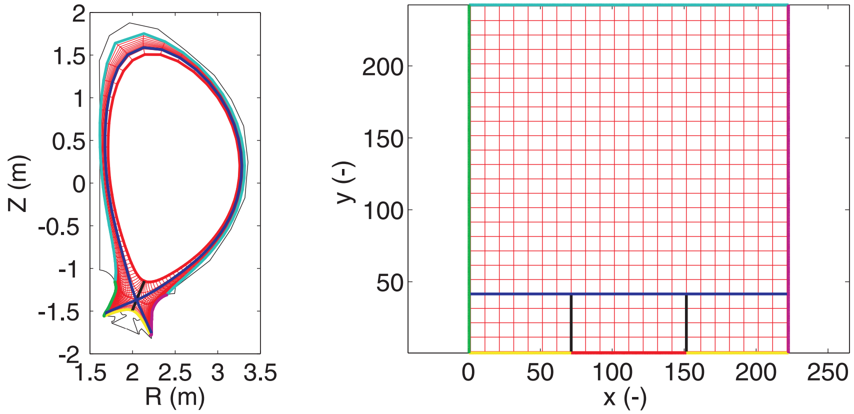
Since the mathematics in deriving the individual equations and boundary conditions does not contribute to the further understanding, the derivation is moved to Appendix B. In the appendix, some further remarks on interesting cost functionals defined on the domain and boundaries are also given.

3.4 Implementation Aspects

In order to solve the state and adjoint equations presented in the previous sections, a finite volume solver has been developed in this thesis. In Section 3.4.1, a brief overview of the most important numerical aspects of the state solver is given. Since the aim is to assess the use of adjoint sensitivity methods for edge plasma applications — in particular for later use with B2(-EIRENE) — the state solver uses numerical methods and iterative schemes closely resembling those of B2, originally developed by Braams [14]. A detailed treatment of the B2 plasma fluid solver can be found in Ref. [5]. Differences between the solver developed in this thesis and B2 will be highlighted. Next, Section 3.4.2 focuses on the adjoint solver. Similarities and differences with the state solver are discussed.

3.4.1 Implementation of the State Equations

The state equations are discretized with a finite volume method on a two dimensional grid with quadrilateral cells. The third spatial direction is assumed to be an ignorable coordinate or a symmetry direction. For tokamak geometry, the grid lies in the poloidal plane, while the toroidal direction is the symmetry direction, see Fig. 3.3 (a) for a typical grid. In order to describe a divertor configuration with a magnetic X-point on a topologically rectangular numerical grid, so-called *cuts* are introduced, see Fig. 3.3 (b). The cuts are represented by the thick black lines in the figure.



(a) Physical grid

(b) Numerical grid

Figure 3.3: (a) Typical grid for an edge plasma simulation (scaled ITER F57 equilibrium [75], see Chapter 5, Section 5.3). In order to improve visibility, not all grid lines are shown. The thick blue line is the separatrix, the thick black lines represent the cuts. Colors indicate the different boundaries. (b) Topologically rectangular numerical grid.

The anisotropy in transport requires special care in the discretization to avoid the contamination of radial transport due to discretization errors. To this end, field aligned grids are used to strictly separate the poloidal and radial transport processes numerically. In practice, this means two of the cell faces have to be aligned with the magnetic flux surfaces (i.e. with the poloidal coordinate lines). In this way, there is only radial flow of particles, momentum and energy across these faces. For numerical accuracy, it is desirable to keep the other two faces as close to orthogonal as possible (i.e. aligned with the radial coordinate lines). However, these faces can be distorted to match the divertor targets, which are usually strongly inclined with respect to the poloidal magnetic field to increase the plasma-wetted area.

A staggered grid configuration is used for the state variables. Scalar quantities (n, p_{n}, T,...) are defined in cell centers, while vector quantities ($u_{||}$, u_r,...) are stored on the cell faces.

Since the state equations are highly nonlinear, an iterative procedure is required to solve them. This iterative procedure resembles the one used in B2. The different plasma parameters are updated one after the other by solving correction equations. During the solution of one equation, it is assumed that other state variables are known. The following update scheme is used:

1. Start from an initial state $\mathbf{q}_0$.
2. Solve the neutral pressure diffusion equation to update the neutral density and pressure.
3. Solve the parallel momentum equation to update the parallel velocity.
4. Solve a pressure-correction equation to update ion density and parallel velocity simultaneously.
5. Solve the energy equation to update the temperature.
6. Iterate steps 2–5 until convergence.

Every step in the iterative procedure involves the solution of a convection diffusion type equation. The pressure-correction equation is discussed in the next section. Then, the text will zoom in on the general numerical treatment of a convection diffusion equation.

Pressure-correction equation

As in B2, a pressure-correction equation is used to enforce ion continuity. The pressure-correction equation can be seen as an extension of the SIMPLE method for incompressible flows to flows with arbitrary Mach number [5, 34]. This is especially useful for the edge plasma, because the flow typically accelerates from Mach number almost zero upstream to sonic speed at the sheath entrance.

The starting point in the derivation of the pressure-correction equation is the linearized continuity equation,

$$\frac{\partial \delta n}{\partial t} + \frac{1}{\sqrt{g}}\frac{\partial}{\partial \theta}\left(\frac{\sqrt{g}}{h_\theta}(\delta n u_\theta + n b_\theta \delta u_{||})\right) - \frac{1}{\sqrt{g}}\frac{\partial}{\partial r}\left(\frac{\sqrt{g}}{h_r^2}D^{\mathrm{i}}\frac{\partial \delta n}{\partial r}\right) - \frac{\partial S_{n_\mathrm{i}}}{\partial n}\delta n = R,$$

with

$$R = S_{n_\mathrm{i}} - \frac{1}{\sqrt{g}}\frac{\partial}{\partial \theta}\left(\frac{\sqrt{g}}{h_\theta} n u_\theta\right) + \frac{1}{\sqrt{g}}\frac{\partial}{\partial r}\left(\frac{\sqrt{g}}{h_r^2}D^{\mathrm{i}}\frac{\partial n}{\partial r}\right).$$

Assuming the temperature is constant during the update of the continuity equation, the equation of state $p = 2nT$ can be used to relate density changes to pressure changes: $\delta p = 2T\delta n$. Since the pressure is the main driving force in the parallel momentum equation, this equation can be used to relate pressure changes to parallel velocity changes, $\delta u_{||} \approx -c_\theta \frac{b_\theta}{h_\theta}\frac{\partial \delta p}{\partial \theta}$. Patankar [90] suggests to take $c_\theta = 1/A_P$, with A_P the coefficient on the main diagonal of the discretized momentum equation. The treatment of radial terms is somewhat different than in B2. Since there is no coupling with the radial pressure gradient through a radial momentum equation, the diffusion coefficient is merely scaled with $1/2T$ to relate density-gradient to pressure-gradient changes. The resulting pressure-correction equation is

$$\begin{aligned} \frac{\partial}{\partial t}\left(\frac{\delta p}{2T}\right) + \frac{1}{\sqrt{g}}\frac{\partial}{\partial \theta}\left(\frac{\sqrt{g}}{h_\theta}\left(u_\theta \frac{\delta p}{2T} - c_\theta n b_\theta^2 \frac{1}{h_\theta}\frac{\partial \delta p}{\partial \theta}\right)\right) & \\ - \frac{1}{\sqrt{g}}\frac{\partial}{\partial r}\left(\frac{\sqrt{g}}{h_r^2}\frac{D^{\mathrm{i}}}{2T}\frac{\partial \delta p}{\partial r}\right) - \frac{\partial S_{n_\mathrm{i}}}{\partial n}\frac{\delta n}{2T} &= R. \end{aligned} \tag{3.53}$$

Finite volume discretization of a convection-diffusion equation

At the heart of the code is a finite volume solver for convection-diffusion equations of the form

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\mathbf{C}\phi - D\nabla\phi) = S, \tag{3.54}$$

with $\mathbf{C} = (C^\theta, C^r)^T$ and $D = \mathrm{diag}(D^\theta, D^r)$ defining the anisotropic poloidal and radial convective and diffusive transport of the parameter ϕ. The source $S = S_\mathrm{c} + S_\mathrm{v}\phi$ is split in a constant part S_c and a variable part $S_\mathrm{v}\phi$. When solving the equation for ϕ, it is assumed that all coefficients in the equation are known (for example, from a previous iteration).

In a finite volume approach, equation (3.54) is first integrated over a cell Ω. Using the divergence theorem, the equation becomes

$$\frac{\partial}{\partial t}\int_\Omega \phi \,\mathrm{d}\boldsymbol{\omega} + \int_\Sigma (\mathbf{C}\phi - D\nabla\phi)\cdot \boldsymbol{\nu}\,\mathrm{d}\sigma = \int_\Omega S\,\mathrm{d}\boldsymbol{\omega}. \tag{3.55}$$

Σ is the surface enclosing the cell, and $\boldsymbol{\nu}$ the unit normal pointing outward. The volume integrals are discretized by assuming that cell center values of ϕ represent volume averages,

$$\begin{aligned} \int_\Omega \phi\,\mathrm{d}\boldsymbol{\omega} &\approx \phi_P \Omega, \\ \int_\Omega S\,\mathrm{d}\boldsymbol{\omega} &\approx (S_\mathrm{c} + S_\mathrm{v}\phi_P)\Omega. \end{aligned}$$

The surface integral in (3.55) is a sum over the individual faces of a cell. In case of an orthogonal grid, Fig. 3.4, the normal to the surface coincides with either the poloidal or radial direction. Therefore, there is only a poloidal or a radial flux through a cell face. For example, fluxes through east and north cell faces are discretized as

$$\begin{aligned} F_e &\approx \left(C^{\theta}\phi - D^{\theta}\nabla_{\theta}\phi\right)_e \Sigma_e, \\ F_n &\approx \left(C^{r}\phi - D^{r}\nabla_{r}\phi\right)_n \Sigma_n. \end{aligned}$$

The face values of ϕ, $\nabla_{\theta}\phi$ and $\nabla_r\phi$ can be approximated based on the five-point stencil depicted in Fig. 3.4. Different interpolation schemes can be used:

- Upwind interpolation
- Linear interpolation
- Hybrid interpolation

The hybrid scheme is based on an approximation to the solution of a one dimensional convection diffusion equation with constant convective and diffusive coefficients [90]. The upwind scheme is used for the poloidal direction in the ion continuity equation. Since the convective term in the neutral continuity equation bears a large similarity with the convective term in the ion continuity equation, it is also discretized with the upwind scheme. For the momentum and energy equations, both linear and hybrid schemes are used.

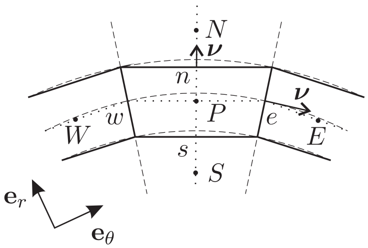


Figure 3.4: Five-point discretization stencil for orthogonal grids. The dashed lines represent the poloidal and radial coordinate lines. The solid lines are their discretized counterparts, i.e. the cell faces. Dotted lines are the connector lines between cell centers, which also follow (discretized) poloidal and radial coordinate lines.

In case the ‘poloidal’ faces are not perpendicular to the poloidal projection of the magnetic field, the grid cells no longer form an orthogonal coordinate system. This has an influence on the discretization of both poloidal and radial fluxes, which then require a more elaborate nine-point stencil for their proper evaluation, see Fig. 3.5. Indeed, the face fluxes are now

$$\begin{aligned} F_e &\approx \left(\left(C^\theta\phi - D^\theta\nabla_\theta\phi\right)\nu_\theta + \left(C^r\phi - D^r\nabla_r\phi\right)\nu_r\right)_e \Sigma_e, && (3.56)\\ F_n &\approx \left(C^r\phi - D^r\nabla_r\phi\right)_n \Sigma_n. && (3.57)\end{aligned}$$

For the poloidal faces, both poloidal and radial flows across the surface have to be evaluated, Eq. (3.56). The poloidal component can be treated in the same fashion as on orthogonal grids, since the connector line between the cells is still in the poloidal direction. The only difference is that the poloidal projection of the face area $\nu_\theta\Sigma_e$ is needed. Evaluating the radial flow across a poloidal face is more complicated. For the convective part, ϕ_e can still be determined based on linear interpolation between cells P and E. The radial component of the gradient of ϕ, on the other hand, is not readily available. It can be constructed based on the poloidal and the tangential components of the gradient. This can be seen by expanding the gradient in two different (local) orthogonal coordinate systems (see Fig. 3.5):

$$\begin{aligned} \nabla\phi &= (\nabla_\theta\phi)\mathbf{e}_\theta + (\nabla_r\phi)\mathbf{e}_r,\\ \nabla\phi &= (\nabla_{\boldsymbol{\nu}}\phi)\boldsymbol{\nu} + (\nabla_{\boldsymbol{\tau}}\phi)\boldsymbol{\tau}.\end{aligned}$$

Taking the dot product of the second expression with $\mathbf{e}_r$ and eliminating $\nabla_{\boldsymbol{\nu}}\phi$ by taking the dot product of the first expression with $\boldsymbol{\nu}$ gives

$$\nabla_r\phi = (\nabla_\theta\phi)\frac{\boldsymbol{\nu}\cdot\mathbf{e}_r}{\boldsymbol{\nu}\cdot\mathbf{e}_\theta} + (\nabla_{\boldsymbol{\tau}}\phi)\frac{1}{\boldsymbol{\nu}\cdot\mathbf{e}_\theta}. \tag{3.58}$$

The poloidal component of the gradient can be computed easily. For the tangential component, the values of ϕ at the cell vertices are needed, which in turn can be found by averaging of the surrounding cell values. This leads to the nine-point stencil.

For the radial faces, Eq. (3.57) shows that the radial component of the gradient is needed. However, the distortion of poloidal cell faces also means that the connector line between the cell centers of a cell and its northern neighbor is no longer aligned with the radial direction. Again, the radial component of the gradient can be constructed by using the gradient along the connector line and the gradient tangential to the surface. To this end, write

$$\begin{aligned} \nabla\phi &= (\nabla_\theta\phi)\mathbf{e}_\theta + (\nabla_r\phi)\mathbf{e}_r,\\ \nabla\phi &= (\nabla_{\boldsymbol{\rho}}\phi)\boldsymbol{\rho} + (\nabla_{\boldsymbol{\tau}}\phi)\boldsymbol{\tau}.\end{aligned}$$

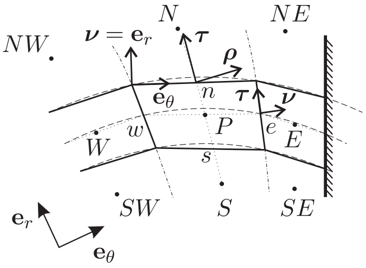


Figure 3.5: Nine-point discretization stencil for non-orthogonal grids. Two cell faces remain aligned with the magnetic field, two are distorted to match the surface. The dashed lines represent the poloidal coordinate lines. Dash-dotted lines are distorted 'radial' lines. The solid lines are the cell faces. Dotted lines are the connector lines between cell centers. At each face, two coordinate systems are defined: a normal-tangential system, and a system along and perpendicular to the connector line between cell centers.

Combining these equations in a similar way as for the poloidal faces gives

$$\nabla_r \phi = (\nabla_{\boldsymbol{\tau}} \phi) \frac{1}{\boldsymbol{\tau} \cdot \mathbf{e}_r} + (\nabla_\theta \phi) \frac{\boldsymbol{\rho} \cdot \mathbf{e}_r}{\boldsymbol{\rho} \cdot \mathbf{e}_\theta}. \tag{3.59}$$

The first term in this equation is the 'expected' contribution. The second term is a correction which takes into account the non-orthogonality of the cells. Again, it is this second term which requires the vertex values of ϕ and leads to the nine-point stencil. Remark that the nine-point flux expressions for distorted meshes nicely reduce to the corresponding five-point expressions if the grid is orthogonal. If the grid is not too distorted, the corrections are expected to be small.

In the B2 code, it is assumed that all grid cells are orthogonal, even though the actual grids provided may be distorted to match the divertor targets. As a consequence, the radial flows across poloidal cell faces in Eq. (3.56) are neglected, as well as the correction to the radial flow at radial cell faces, i.e. the second term in Eq. (3.59). For the plasma component, the former approximation can be justified by the highly anisotropic transport. Indeed, the radial flow is expected to be at least an order of magnitude smaller than the poloidal flow. However, this is not true for the neutrals and the corresponding neutral energy transport, so this term must be kept here. Neglecting the correction to the radial transport at radial cell faces seems harder to defend, especially since large gradients tend to develop towards the targets, which are now not taken into account. The influence of neglecting this correction remains to be quantified for B2 simulations.

Setup of the correction equation

The aim is to solve the steady-state form of Eq. (3.55) by making the residual R, defined as

$$R = S - \nabla \cdot (\mathbf{C}\phi - D\nabla\phi), \tag{3.60}$$

zero. This can be achieved by using the time derivative to iterate towards steady state with a so-called *false-time-stepping* procedure. Since time step restrictions for explicit time integration are very strict, implicit time integration is chosen. Linearizing equation (3.54) gives

$$\frac{\partial \Delta\phi}{\partial t} + \nabla \cdot \left(\left(\mathbf{C} + \frac{\partial \mathbf{C}}{\partial \phi}\phi \right) \Delta\phi - D\nabla(\Delta\phi) \right) - S_{\mathrm{v}}\Delta\phi = R, \tag{3.61}$$

which can now be solved for the correction $\Delta\phi$. Additional under relaxation is performed by updating ϕ as $\phi + \alpha\Delta\phi$, with $0 < \alpha < 1$.

The discretization of the right hand side of Eq. (3.61) has been discussed in the previous paragraphs. Discretization of the left hand side leads to the matrix which has to be inverted. The same discretization schemes as on the right hand side could be used for the linearized fluxes, but this is not necessary. In order to limit the bandwidth of the matrix — and thus to reduce the computational time needed for a single matrix inversion — it can be chosen to work with a five-point stencil rather than with a nine-point stencil in the correction equation. In practice, this means the second terms on the right hand sides of equations (3.58) and (3.59) are not included in the stencil of the correction equation. An iterative scheme based on a five-point stencil updates is expected to be sufficiently robust in case grid orthogonality is not too strong, and if radial transport is not too dominant. In those cases, the terms which are neglected in the correction equation are small. However, in this work it has been observed that especially for strongly distorted grids and at high recycling conditions, the five-point update scheme is unstable. Therefore, also a full nine-point stencil correction scheme is implemented.

Boundary conditions

Boundary conditions are imposed through guard cells, in the same way as in the B2 code. The guard cells are infinitesimal cells placed around the grid, see Fig. 3.6. Due to the small size of the guard cell in the direction perpendicular to the domain, the only non negligible terms in the equation of the guard cell are the flux coming from the domain and the source term:

$$F \equiv (\mathbf{C}\phi - D\nabla\phi)_f \cdot \boldsymbol{\nu}\Sigma_f = (S_{\mathrm{c}} + S_{\mathrm{v}}\phi_P)\,\Omega. \tag{3.62}$$

The source term can be manipulated to impose a certain boundary condition:

- A flux $F = F_{\mathrm{d}}$ can be directly imposed by choosing the source coefficients to have $F_{\mathrm{d}} = (S_{\mathrm{c}} + S_{\mathrm{v}}\phi_P)\,\Omega$.
- A Dirichlet condition $\phi = \phi_0$ is obtained by choosing $S_{\mathrm{c}}\Omega = B\phi_0$ and $S_{\mathrm{v}}\Omega = -B$, with B a sufficiently large number so that the fluxes on the left hand side of (3.62) are numerically negligible.
- More complex mixed conditions can be obtained in a similar way by proper choice of the sources in the guard cells.

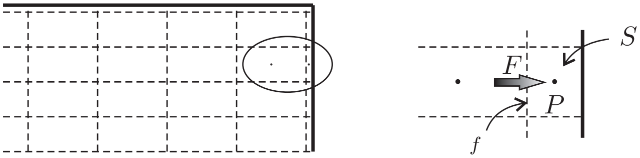


Figure 3.6: Infinitesimal guard cells surround the domain to impose boundary conditions. Figure based on Ref. [5].

3.4.2 Implementation of the Adjoint Equations

The solution of the adjoint equations requires some further consideration. First of all, it is stressed that the adjoint equations are linear. Therefore, in principle they could be solved with exactly one matrix inversion for the coupled set of adjoint equations and boundary conditions. However, due to the complex coupling in the adjoint equations and boundary conditions, discretizing the coupled equations exactly is quite a complex task.

On the other hand, since a state solver is already available, it is interesting to investigate how much of the numerical routines developed for this solver can be recuperated in the adjoint solver. Reusing computational routines not only makes the implementation less error prone, it also allows to add an adjoint solver to an existing code as B2 at a relatively low implementation cost. Therefore, the adjoint equations are relaxed one by one through the use of correction equations. Keeping to the ‘adjoint idea’, the order of the updates is reversed compared to the forward equations:

1. Start from an initial guess $\mathbf{q}_0^*$.
2. Solve the adjoint energy equation to update the adjoint temperature.

3. Solve the adjoint continuity equation to update adjoint ion density. Alternatively, solve the adjoint pressure-correction equation for coupled adjoint density and velocity updates.
4. Solve the adjoint parallel momentum equation to update the adjoint parallel velocity.
5. Solve the adjoint neutral pressure diffusion equation to update the adjoint neutral pressure.
6. Iterate steps 2–5 until convergence.

By writing the adjoint equations in conservative form (3.49), it has already been indicated that their structure is very similar to that of the state equations. After decoupling the adjoint equations, Eqs. (B.5)–(B.8), it is seen that an adjoint equation contains identical diffusive terms as the corresponding state equation, Eqs. (3.30)–(3.33), and convective terms with switched sign. In view of the general convection diffusion equation presented above, the adjoint equations to be solved are of the form

$$\frac{\partial \phi^*}{\partial t} - \nabla \cdot (\mathbf{C}\phi^* + D\nabla\phi^*) = S^*. \tag{3.63}$$

The source term S^* has a complex dependency on the (other) adjoint variables. In particular, the source term in the equation of one particular adjoint variable may contain several convection-diffusion type terms due to the other adjoint variables. These sources are treated explicitly in the iterative procedure. Discretization of equation (3.63) can proceed in analogous way to the state equations, making it particularly easy to reuse parts of the code. An important difference is that the coefficients $\mathbf{C}$ and D in the adjoint equation depend on the state variables only (not on the adjoint variables), and therefore the matrices have to be assembled only once.

The time-dependent term in Eq. (3.63) is introduced for under relaxation. Note that when time-dependent problems are studied, the time derivative term in the adjoint equations typically has a negative sign. In order to have a stable problem, the adjoint equations then have to be integrated backwards in time (i.e. with a negative time step). When studying steady-state problems, as is the case in the present work, it is customary to use the ‘positive’ time derivative term as in Eq. (3.63), and integrate the adjoint equations ‘forward’ in time. As the nature of the adjoint equations resembles that of the state equations, also the time constants are expected to be similar. Therefore, in principle the adjoint equations converge in roughly the same time as the state equations. If the discrete adjoint approach is used, the asymptotic convergence behavior can be made identical, see for example Ref. [45]. When using the continuous adjoint approach, this can also be achieved by making the adjoint iterative scheme resemble as closely as possible the dual of the forward iterative scheme. Specifically, the introduction of an adjoint pressure-correction equation seems necessary to guarantee the same convergence behavior.

Adjoint pressure-correction equation

When using the adjoint ion continuity equation in the iterative procedure to update the ion density, it has been found that for strongly coupled problems stability requires the time step of the adjoint equations to be reduced compared to the forward equations. This is an issue for the performance of the entire optimization algorithm, because it increases the number of iterations needed for convergence.

A cure to this problem can be constructed based on the solution scheme of the forward equations. There, the updates of the continuity equation are obtained from a pressure-correction equation. In the linearized poloidal particle flux,

$$\delta(nu_\theta) = u_\theta \delta n + n \delta u_\theta, \tag{3.64}$$

depending on the local compressibility either the term involving density variations (at the target) or the one involving velocity variations (upstream) is dominant in absorbing particle flow fluctuations. By keeping both terms in the update equation, the pressure-correction equation (3.53) was obtained, which gives more robust coupled velocity and density updates.

Similarly, it can be expected that keeping these two terms coupled in the adjoint equations can improve performance of the iterative scheme. In the adjoint equations, the first term on the right hand side of Eq. (3.64) is the convective term in the adjoint ion continuity equation. The second term is found in the parallel momentum equation. Therefore, based on the adjoint parallel momentum equation, the simplified relation

$$u_{||}^* \approx \frac{n}{A_P^*} \frac{b_\theta}{h_\theta} \frac{\partial n^*}{\partial \theta}$$

is proposed. A_P^* is the coefficient on the main diagonal of the momentum equation. By inserting this into the adjoint ion continuity equation (B.5), the update equation becomes

$$\begin{aligned} &- \frac{1}{\sqrt{g}} \frac{\partial}{\partial \theta} \left(\frac{\sqrt{g}}{h_\theta} \left(u_\theta n^* + (m u_\theta u_{||} + 2 b_\theta T) \frac{n}{A_P^*} \frac{b_\theta}{h_\theta} \frac{\partial n^*}{\partial \theta} \right) \right) \\ &\qquad - \frac{1}{\sqrt{g}} \frac{\partial}{\partial r} \left(\frac{\sqrt{g}}{h_r^2} \left(D^{\mathrm{i}} \frac{\partial n^*}{\partial r} \right) \right) = S_{n_\mathrm{i}}^* - n^* \frac{1}{\sqrt{g}} \frac{\partial}{\partial \theta} \left(\frac{\sqrt{g}}{h_\theta} u_\theta \right) + \frac{\partial J_1}{\partial n}. \end{aligned} \tag{3.65}$$

In this equation, only the terms which are updated after solving the equation are written out explicitly. All other terms are absorbed in the source $S^*_{n_\mathrm{i}}$. By introducing this 'adjoint pressure-correction equation', the same time step as in the forward equations is allowed again.

3.5 Optimization Algorithms

The third and final part in a design code is the optimization algorithm. This algorithm is responsible for finding a solution satisfying the necessary optimality conditions (2.36). Optimization in its own is a vast research domain, with many different problem types and both general purpose or tailored methods. However, in optimization problems constrained by PDEs, the nonconvexity, large number of design variables and computational complexity brought by the PDE constraint often render only local methods which rely on cost functional evaluations and first order sensitivity information viable. For interesting discussions on optimization algorithms in the context of PDE constraints, see a.o. Refs. [52, 12, 13]. Below, some black-box methods and a one-shot approach are discussed.

For certain PDE constrained problems, derivative free methods such as genetic algorithms are used, see e.g. Ref. [126]. However, when the number of design variables becomes large, the total number of cost functional evaluations performed by these algorithms may become prohibitive. Although the number of evaluations can be minimized by using for example *metamodels* [67], possibly based on inexact state solutions [68] or gradient information [44], the cost would remain unacceptable for edge plasma applications. Therefore, these methods are not considered here.

3.5.1 Black-Box Methods

Black-box methods work directly on the reduced cost functional $\hat{J}(\phi)$. Depending on the algorithm, evaluations of the cost functional and design sensitivities are required, and possibly higher order derivatives (the Hessian). The optimization problem is solved without a specific reference to the underlying PDE. The main influence of the PDE is that cost functional evaluations are expensive. This is especially true for the edge plasma applications considered in this thesis. With the adjoint method, the evaluation of first order sensitivities can be done at roughly the same cost as a cost functional evaluation. However, it is not realistic to assume that higher order sensitivity information is readily available. Therefore, two gradient-based methods will be considered which require only the first order (shape) derivative:

- Steepest Descent method
- BFGS method (a quasi-Newton method)

Below, a brief presentation of the methods and their most relevant characteristics for the current application is given. For details, see for example Ref. [87]. Both methods start from an initial design ϕ_0, and generate a sequence of iterates ϕ_k until an optimum is found. At each iteration, a search direction $\delta\phi_k$ is computed, after which an inexact line search is performed in order to obtain a step length τ_k which approximately minimizes the reduced cost functional $\hat{J}(\phi_k + \tau\delta\phi_k)$ in the direction $\delta\phi_k$. For a schematic representation of the algorithms, see Fig. 3.7 (a).

The steepest descent method

The steepest descent method is conceptually the easiest method. The search direction is the direction of the negative gradient,

$$\delta\phi_k = -\nabla\hat{J}(\phi_k),$$

which locally gives the fastest decrease in cost functional. In order to ensure convergence of the algorithm to a (local) minimum, a line search on the value of the cost functional is needed. The line search algorithm used in this study is a simple backtracking algorithm, where a step length is determined which satisfies the sufficient decrease condition or *Armijo condition*:

$$\hat{J}(\phi_k + \tau_k\delta\phi_k) \leq \hat{J}(\phi_k) + c_1\tau_k \left(\nabla\hat{J}(\phi_k), \delta\phi_k\right), \tag{3.66}$$

with c_1 a (small) constant.

It is well known that the steepest descent method suffers from slow linear convergence, and is very sensitive to problem scaling [87]. In particular, it is known that aerodynamic shape design problems are often ill-conditioned, see for example Refs. [3, 108]. In shape optimization problems, on the other hand, performance can be quite acceptable due to the gradient smoothing. Indeed, it is often found that the Sobolev smoothing introduced in Section 2.4.2 makes the reduced Hessian close to identity [13]. As a result, the problem is well scaled, and the steepest descent direction resembles the Newton direction.

The fact that a line search is needed to ensure theoretical convergence of the algorithm can be an issue when using a continuous adjoint method. Due to discretization errors, there will be a discrepancy between the derivatives computed by the continuous adjoint method and the actual derivatives of the discretized optimization problem. Although the differences should decrease with grid refinement, in practical problems sooner or later discretization errors will start to dominate, and the shape derivative computed with the continuous adjoint method may no longer provide a descent direction for the discretized optimization problem. In this case, the line search will stall and the optimization algorithm has to be interrupted. Some authors have successfully used the steepest descent method in combination with the continuous adjoint method by monitoring the norm of the gradient rather than the value of the cost functional, sometimes even eliminating the line search altogether. For example, in Ref. [65] the steepest descent method is used with a fixed step length. This approach can be interpreted as explicit Euler integration of the equation

$$\frac{\partial \phi}{\partial \tau} = -\nabla \hat{J}(\phi).$$

In Ref. [65], it is observed that this time-stepping procedure works very well even if $\nabla \hat{J}(\phi)$ is computed with only partially converged solutions to the state and adjoint equations. By eliminating the line search and reducing the number of iterations of the solvers, this approach closely resembles the one-shot method which will be discussed below.

The BFGS method

Convergence speed of the optimization method can be increased by using higher order derivatives. Ideally, the Newton method is used, which provides quadratic convergence properties close to the optimum. In the Newton method, the search direction is

$$\delta\phi_k = -\hat{J}_{\phi\phi,k}^{-1} \nabla \hat{J}(\phi_k),$$

with $\hat{J}_{\phi\phi,k}$ the Hessian evaluated at iterate ϕ_k.

In case the Hessian is not available, quasi-Newton methods are an interesting alternative. Instead of using the true Hessian, these methods iteratively construct approximations to the Hessian or its inverse by measuring changes in first order derivatives, thereby gradually collecting information about the optimum. Convergence is typically superlinear. A method which has proven very successful in practice is the BFGS method, in which the approximation H_k to the inverse Hessian is iteratively updated by [87]

$$H_{k+1} = (I - \rho_k s_k y_k^T) H_k (I - \rho_k y_k s_k^T) + \rho_k s_k s_k^T,$$

with

$$s_k = \phi_{k+1} - \phi_k, \qquad y_k = \nabla \hat{J}_{k+1} - \nabla \hat{J}_k, \qquad \rho_k = \frac{1}{y_k^T s_k}.$$

This update is constructed based on a local quadratic model of the objective, in such way that the gradient at iterates k and $k+1$ are correctly reproduced. In order to keep H_{k+1} symmetric positive definite, the curvature condition $s_k^T y_k > 0$ must hold. This can be ensured by requiring the line search algorithm to satisfy the (strong) Wolfe conditions [87]. The Wolfe conditions combine the Armijo rule given in Eq. (3.66) with the following curvature condition on the gradient:

$$\left|\left(\nabla \hat{J}(\phi_k + \tau_k \delta\phi_k), \delta\phi_k\right)\right| \leq c_2 \left|\left(\nabla \hat{J}(\phi_k), \delta\phi_k\right)\right|, \tag{3.67}$$

with $0 < c_1 < c_2 < 1$. Values $c_1 = 10^{-4}$ and $c_2 = 0.9$ are recommended in Ref. [87].

In the initial iteration, an estimate of the Hessian has to be provided. Although the robustness of the BFGS algorithm may depend on this initizalization [87], there is unfortunately no magic formula to do this. Therefore, initially a simple steepest descent step is taken in the direction of the smoothed gradient, and the Hessian is initiated to a multiple of the identity matrix. Since smoothing the gradient tends to approximate the action of the Hessian in shape optimization problems, the identity matrix may be expected to work quite well as initial guess.

3.5.2 One-Shot Method

The term *one-shot method* refers to an approach where the state and adjoint equations are not solved exactly during an optimization iteration [119]. Instead state, adjoint and design equations are gradually relaxed during the optimization run towards the optimal solution, and are only feasible at the end. Over the years, several authors have used different variants of the one-shot method. Here, the approach of Refs. [55, 57] is followed most closely.

In Ref. [55], an interpretation of a one-shot method or *simultaneous pseudo-timestepping* method is given in terms of the iterative solution of the optimality system (2.36). Gradient-based methods as described above can be thought of as marching on the system

$$\left\{\begin{array}{rcl} \nabla_{\mathbf{q}^*} L(\phi, \mathbf{q}, \mathbf{q}^*) & = & 0, \\ \nabla_{\mathbf{q}}\, L(\phi, \mathbf{q}, \mathbf{q}^*) & = & 0, \\ \frac{\partial \phi}{\partial t} + \nabla_{\phi}\, L(\phi, \mathbf{q}, \mathbf{q}^*) & = & 0. \end{array}\right.$$

At every iteration, state and adjoint equations are solved exactly, the design sensitivity is evaluated, and a step is taken in design space. However, computing accurate state and ajoint solutions is expensive, and thus it may be advantageous to partially relax the requirement of a feasible state and adjoint trajectory.

Therefore, Ta'asan [120, 119] originally suggested to iterate on state and adjoint equations, while keeping the design equation as an additional boundary condition,

$$\left\{ \begin{array}{rcl} \frac{\partial \mathbf{q}}{\partial t} + \nabla_{\mathbf{q}^*} L(\phi, \mathbf{q}, \mathbf{q}^*) & = & 0, \\ \frac{\partial \mathbf{q}^*}{\partial t} + \nabla_{\mathbf{q}} \ L(\phi, \mathbf{q}, \mathbf{q}^*) & = & 0, \\ \nabla_{\phi} \ L(\phi, \mathbf{q}, \mathbf{q}^*) & = & 0. \end{array} \right.$$

This idea has been further generalized in Ref. [55] towards the complete iterative solution of the optimality system,

$$\left\{ \begin{array}{rcl} \frac{\partial \mathbf{q}}{\partial t} + \nabla_{\mathbf{q}^*} L(\phi, \mathbf{q}, \mathbf{q}^*) & = & 0, \\ \frac{\partial \mathbf{q}^*}{\partial t} + \nabla_{\mathbf{q}} \ L(\phi, \mathbf{q}, \mathbf{q}^*) & = & 0, \\ \frac{\partial \phi}{\partial t} + \nabla_{\phi} \ L(\phi, \mathbf{q}, \mathbf{q}^*) & = & 0. \end{array} \right.$$

In Refs. [55, 13], one-shot methods are interpreted in view of reduced Sequential Quadractic Programming (rSQP) methods, which perform iterations

$$\begin{pmatrix} 0 & 0 & \boldsymbol{\mathcal{B}}_{\mathbf{q}}^* \\ 0 & \hat{J}_{\phi\phi} & \boldsymbol{\mathcal{B}}_{\phi}^* \\ \boldsymbol{\mathcal{B}}_{\mathbf{q}} & \boldsymbol{\mathcal{B}}_{\phi} & 0 \end{pmatrix} \begin{pmatrix} \Delta \mathbf{q} \\ \Delta \phi \\ \mathbf{q}^* \end{pmatrix} = \begin{pmatrix} -\nabla_{\mathbf{q}} J \\ -\nabla_{\phi} J \\ \boldsymbol{\mathcal{B}} \end{pmatrix} \tag{3.68}$$

on the optimization problem. Reduced SQP methods exploit the problem structure by using separability between control and state variables to see the constraint $\boldsymbol{\mathcal{B}}(\phi, \mathbf{q})$ as an implicit relation for $\mathbf{q}(\phi)$ (i.e. assuming $\boldsymbol{\mathcal{B}}_{\mathbf{q}}$ is invertible), and then work on the reduced cost functional $\hat{J}(\phi)$. The first equation in Eq. (3.68) is the adjoint equation. The second equation is a Newton step on the reduced cost functional, with $\hat{J}_{\phi\phi}$ the Hessian of the reduced cost functional and $\nabla_{\phi} J + \boldsymbol{\mathcal{B}}_{\phi}^* \mathbf{q}^*$ the gradient. Finally, the third equation is a (single) Newton update of the state equation constraint. The fact that the state equations are not solved exactly in each iteration is already a benefit in terms of computational time. In practice, the Hessian $\hat{J}_{\phi\phi}$ will be approximated, for example with a BFGS scheme. Furthermore, exact Newton steps of the state equations are usually not performed, and also $\boldsymbol{\mathcal{B}}_{\mathbf{q}}$ is approximated.

One-shot methods may be seen as an approximate iteration of a rSQP method. Rather than using an (exact or approximate) Newton step on the state equations, one iteration of the (nonlinear) state equations is performed with the updated design variables. Furthermore, the adjoint equations are not solved exactly in each step, but again only one iteration is performed. As a result, the Hessian approximation is based on approximate reduced gradients. In this thesis, the Hessian is approximated by Sobolev smoothing. By rewriting Eq. (2.37) as

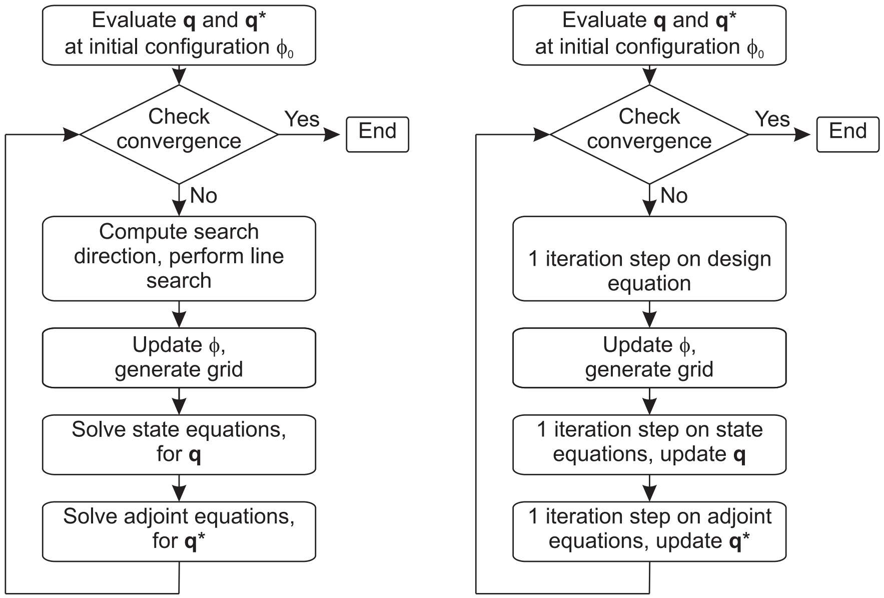
$$\tilde{\mathcal{G}} = (I - \epsilon \Delta_{\Sigma})^{-1} \mathcal{G}$$

with $\mathcal{G}$ the shape gradient, it is seen that Sobolev smoothing implies an approximation $\hat{J}_{\phi\phi} \approx I - \epsilon\Delta_\Sigma$ with resulting approximate Newton direction $\tilde{\mathcal{G}}$.

A schematic representation of the one-shot algorithm is given in Fig. 3.7 (b). A clear advantage of the method is its modularity [13]. The algorithm can make direct use of the state solver *as it is*, and therefore lends itself very well for use with an existing code. This is also apparent in Fig. 3.7, which shows that same code structure can be used as for black-box methods.

3.5.3 Structure of the Optimization Code

To conclude this section, the overall structure of the optimizer is discussed. A schematic representation of the solver is given in given in Fig. 3.7. It has already been pointed out that the solver structure is the same for black-box and one-shot methods, only the numerical parameters differ. Thus, the user can easily switch from a gradient-based to a one-shot loop by tuning a few parameters. In the gradient-based loop (a), state and adjoint equations are solved during each optimization iteration, and a line search is performed in the



(a) Gradient-based loop (b) One-shot loop

Figure 3.7: Schematic representation of optimization loop.

search direction. This line search in itself may again require several (partial) solutions of state and adjoint equations. The number of optimization iterations is relatively small. In the one-shot loop (b), state and adjoint equations are only updated for one or a few iteration steps (not solved), and no line search is performed. Each iteration in the one-shot loop is thus fast compared to the gradient-based loop. However, many iterations are required for convergence.

3.6 Conclusion

In this chapter, the general theory presented in Chapter 2 is elaborated for edge plasma applications. First, the model for the edge plasma is described. The model resembles the models used in edge codes as B2-EIRENE, but makes some additional simplifying assumptions. Most importantly, in order to avoid the complexity related to the kinetic treatment of the neutral component in a first step, a pressure diffusion relation is used for the neutrals instead.

Next, a general derivation of the ajoint equations is given for a system of coupled convection-diffusion equations in curvilinear coordinate systems. The resulting equations are elaborated for the edge plasma model used in this thesis, but can also be applied directly to for example the system of equations of B2.

The edge plasma model and its adjoint are then implemented in a finite volume solver. Details on the solver developed in this thesis are presented. Since an aim of this thesis is to assess the use of the adjoint methodology in larger edge codes — in particular in B2 — the numerical implementation is based on the schemes used in B2. However, some additional features are added such as the nine-point discretization of the fluxes on non-orthogonal grids. Due to the similarity between state and adjoint equations, the numerical algorithms can be used for both systems of equations.

Finally, several optimization algorithms are discussed and the structure of the design code is presented.

Chapter 4

Divertor Shape Optimization in Simplified SOL Geometry

With divertor design formulated as a shape optimization problem in Chapter 2 and the different parts of an optimization code described in Chapter 3, all elements are now available for a thorough assessment of the developed formalism. In order to eliminate some of the complexity related to the curvilinear geometry in a first step, this chapter is built around a simplified 'slab' representation of the plasma edge. A cost functional is defined which aims at peak heat load reduction and optimal spreading of the power load at the targets. This cost functional will reappear throughout the thesis.

The setup of the optimization problem is discussed in Section 4.1. Then, an initial validation of the numerical algorithms and comparison of the different optimization algorithms is presented in Section 4.2. For computational speed, this is performed with a somewhat reduced edge plasma model. Section 4.3.1 focuses more in detail on the shape sensitivities computed with the complete edge plasma model, and important aspects as gradient smoothing and nine-point discretization of the fluxes. Finally, optimized divertor target configurations are presented in Section 4.3.2.

4.1 Setup of the Optimization Problem

4.1.1 Cost Functional and Shape Parametrization

A large part of the thesis will deal with the problem of optimizing the energy flux to the divertor targets by controlling the divertor target shape. To this end, the following cost functional is defined:

$$J(\Omega, \mathbf{q}) = \frac{1}{2} \int_{\mathrm{t}} (Q_{\mathrm{o}} - Q_{\mathrm{d}})^2 \, \mathrm{d}\sigma. \tag{4.1}$$

The integral is taken at the divertor targets. This cost functional measures the square of the difference between the actual energy flux deposited on the target, $Q_{\mathrm{o}} = \mathbf{Q}_{\mathrm{o}} \cdot \boldsymbol{\nu}$, and a desired energy flux, Q_{d}. The subscript 'o' refers to the fact that this energy flux is to be optimized. By choosing Q_{d} as a constant energy flux (well) below the critical flux that can be handled by the target material, the aim is to bring Q_{o} down to acceptable levels, yet still load the target material as uniformly as possible. In this way, optimal use is made of the high-heat-flux components in the divertor.

There are different components to the energy flux deposited at the target. First of all, there is heat Q_{t} convected and conducted by the plasma and neutrals. According to the boundary conditions described in Section 3.1.7, $Q_{\mathrm{t}} = \mathbf{Q}_{\mathrm{t}} \cdot \boldsymbol{\nu} = \delta_{\mathrm{sh}} T \boldsymbol{\Gamma}_{\mathrm{t}} \cdot \boldsymbol{\nu}$, with $\boldsymbol{\Gamma}_{\mathrm{t}} = n u_\theta \mathbf{e}_\theta$. Then, energy is set free at the target due to surface recombination at a rate $Q_{\mathrm{sr}} = E_{\mathrm{p}} \boldsymbol{\Gamma}_{\mathrm{t}} \cdot \boldsymbol{\nu}$, with E_{p} the potential energy of recombination, 13.6 eV for Hydrogen. In principle, the kinetic energy of ions and neutrals should be taken into account as well. Since sound speed is reached at the targets, the kinetic energy flux of the ions scales as $Q_{\mathrm{k}} \sim m n u_\theta u_{||}^2/2 \sim n u_\theta T$, where $u_{||} = c_{\mathrm{s}} = (2T/m)^{1/2}$ has been used. However, the neutrals may take a fraction of this energy back into the plasma as they recycle. Due to the absence of a description for the neutral kinetic energy flux in the model, both ion and neutral contributions are neglected at this point. Electrical currents may also contribute to the energy flux [5], but are neglected in this thesis. Finally, in reactor-scale devices, there can be a significant contribution from radiation. Radiation transport is not considered in the model at this point, but will be treated in Chapter 8. Concluding, in this chapter

$$\begin{aligned} Q_{\mathrm{o}} &= Q_{\mathrm{t}} + Q_{\mathrm{sr}} \\ &= (\delta_{\mathrm{sh}} T + E_{\mathrm{p}})\, n u_\theta \mathbf{e}_\theta \cdot \boldsymbol{\nu}. \end{aligned} \tag{4.2}$$

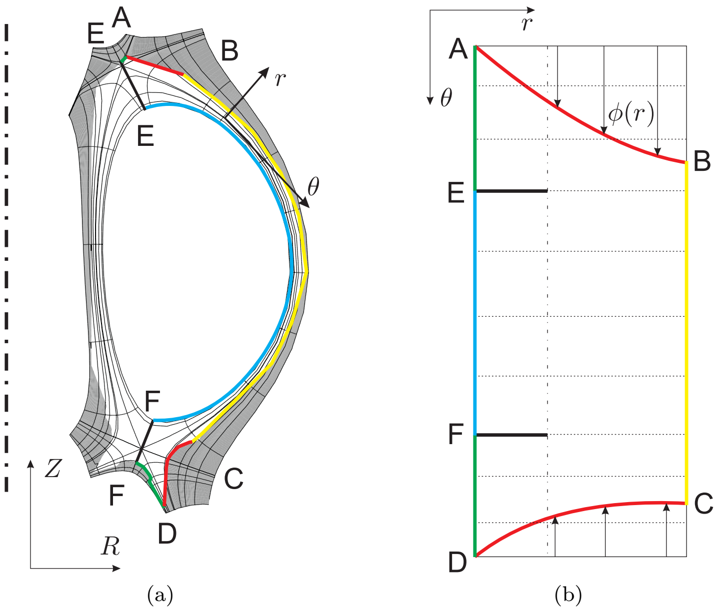


Figure 4.1: (a) Outboard half of ITER divertor and (b) representation of a connected double null divertor in the (θ, r) coordinate system. The dashed line in (b) represents the separatrix.

In this chapter, the SOL will be straightened out to a ‘slab’ or ‘cylindrical shell’ with constant magnetic field pitch b_θ. This means that the poloidal and radial coordinate lines in Fig. 4.1 (b) coincide with the Z and R directions of the fixed cylindrical coordinate system. Such a shell can be interpreted as a simplified, straightened out representation of the outboard half of a connected double null divertor. This geometric simplification allows to have a good grid resolution with a relatively small number of cells, resulting in faster simulations and more manageable numerical validation and grid sensitivity study. Applications in a realistic SOL geometry will be discussed in the next chapter.

The target shape is parametrized by a continuous function $\phi = \phi(r)$ which determines the poloidal coordinate of the target in a poloidal cross-section of the tokamak, see Fig. 4.1. This parametrization allows for a large design freedom, but also ensures that the target will cross each flux surface exactly once, thereby avoiding the need for wide grids. The only restriction on ϕ is the requirement that the vessel remains closed. The latter restriction means that $\phi(A)$, $\phi(B)$, $\phi(C)$, and $\phi(D)$ are fixed during optimization. Translating the parametrization with a function ϕ to the velocity method described in

Chapter 2, the perturbation fields $\boldsymbol{\mathcal{V}}$ are restricted by $\boldsymbol{\mathcal{V}}(\delta\phi)$, where $\delta\phi$ are perturbations to the design variables along the field lines. For ease of notation, the cost functional will be written as

$$J(\phi, \mathbf{q}) \equiv J(\Omega(\phi), \mathbf{q}).$$

The optimal design is then determined by the vector of design variables $\hat{\phi}$ which minimizes the reduced cost functional $\hat{J}(\phi)$.

4.1.2 Shape Sensitivity

According to (2.36), the shape sensitivity of the cost functional is found by taking the partial derivative of the Lagrangian with respect to the geometry. This Lagrangian is

$$L(\phi, \mathbf{q}, \mathbf{q}^*) = J(\phi, \mathbf{q}) + \int_\Omega \mathbf{q}^* \cdot \boldsymbol{\mathcal{B}}(\mathbf{q})\, \mathrm{d}\boldsymbol{\omega} + \int_\Sigma \mathbf{p}^* \cdot \boldsymbol{\mathcal{C}}(\mathbf{q}, \boldsymbol{\nu})\, \mathrm{d}\sigma. \tag{4.3}$$

The state equations $\boldsymbol{\mathcal{B}}(\mathbf{q})$ and boundary conditions $\boldsymbol{\mathcal{C}}(\mathbf{q}, \boldsymbol{\nu})$ were described in Chapter 3, Section 3.1. Using the result from Eq. (2.30), all terms involving the shape derivatives of state and adjoint variables in the shape derivative of the Lagrangian cancel. Furthermore, in the design problem under consideration, only the target boundary is allowed to move. Thus, only (1) the cost functional itself, (2) the domain integral and (3) the boundary conditions at the target contribute to the shape derivative of the reduced cost functional:

$$\dot{\hat{J}} = \dot{L} = \dot{L}_1 + \dot{L}_2 + \dot{L}_3. \tag{4.4}$$

For the cost functional, there is a dependence of the integrand on the surface normal $\boldsymbol{\nu}$, so this functional is of type (2.22). After elaboration of (2.25), the contribution to the shape derivative becomes

$$\begin{aligned} \dot{L}_1 \quad &= \int_\mathrm{t} \left(\frac{1}{2} \left(Q_\mathrm{o} - Q_\mathrm{d}\right)^2 - \left(Q_\mathrm{o} - Q_\mathrm{d}\right) Q_\mathrm{o} \right) \kappa \boldsymbol{\mathcal{V}} \cdot \boldsymbol{\nu}\, \mathrm{d}\sigma \\ &\quad + \int_\mathrm{t} \left(\left(Q_\mathrm{o} - Q_\mathrm{d}\right) \nabla \cdot \mathbf{Q}_\mathrm{o} + \mathbf{Q}_\mathrm{o} \cdot \nabla_\Sigma \left(Q_\mathrm{o} - Q_\mathrm{d}\right)\right) \boldsymbol{\mathcal{V}} \cdot \boldsymbol{\nu}\, \mathrm{d}\sigma. \end{aligned} \tag{4.5}$$

The shape derivative of the domain integral involves evaluating the integrand on the (target) boundary, see Eq. (2.18). If the state equations are satisfied, $\boldsymbol{\mathcal{B}}(\mathbf{q}) = 0$, so

$$\dot{L}_2 = 0. \tag{4.6}$$

The last contribution to the shape derivative stems from

$$\int_{\mathrm{t}} \mathbf{p}^* \cdot \boldsymbol{\mathcal{C}}_{\mathrm{t}} \, \mathrm{d}\sigma,$$

with $\boldsymbol{\mathcal{C}}_{\mathrm{t}}$ from (3.26) and $\mathbf{p}^*$ from (B.11) and (B.12). The boundary conditions for ion continuity, neutral continuity and energy equations are of type (2.21). The boundary condition for the parallel momentum equation is of type (2.20). Therefore, the shape derivative becomes

$$\begin{aligned} \dot{L}_3 \quad = \quad & \int_{\mathrm{t}} \Big(\nabla \cdot \left(n^* \left(\boldsymbol{\Gamma}^{\mathrm{i}} - \boldsymbol{\Gamma}_{\mathrm{t}} \right) \right) + u^*_{||,S} \boldsymbol{\nu} \cdot \nabla \left(\pm c_{\mathrm{s}} - u_{||} \right) \Big) \boldsymbol{\mathcal{V}} \cdot \boldsymbol{\nu} \, \mathrm{d}\sigma \\ & + \int_{\mathrm{t}} \left(\nabla \cdot \left(p^*_{\mathrm{n}} \left(\boldsymbol{\Gamma}^{\mathrm{n}} + R \boldsymbol{\Gamma}_{\mathrm{t}} \right) \right) + \nabla \cdot \left(T^* \left(\mathbf{Q} - \mathbf{Q}_{\mathrm{t}} \right) \right) \right) \boldsymbol{\mathcal{V}} \cdot \boldsymbol{\nu} \, \mathrm{d}\sigma. \end{aligned} \tag{4.7}$$

Discretization of domain and design variables

Poloidal coordinate lines are kept parallel to the magnetic field, but radial lines are distorted to match the target surface. Due to the strong gradients developing at the sheath, the grid needs to be refined significantly towards the targets. Also at the X-point, where there is a singularity in the solution, sufficiently fine cells are needed. Upstream, on the other hand, profiles are expected to vary only slowly with the poloidal coordinate, so relatively large cells can be used. The actual grids used for the test problems will be shown later in this chapter.

In a discretized domain with $N_\theta \times N_r$ cells, the coordinates of the cell vertices lying on the target surface are the control variables. Since the first and last vertices are fixed by the requirement to match dome and main chamber (i.e point A, B, C, and D in Fig. 4.1 are fixed), the number of discrete design variables — and thus the number of simulations required for a single derivative evaluation using finite differences — is $2 \times (N_r - 1)$.

In order to update the design, shape sensitivity information has to be evaluated at the location of the control variables, i.e. at cell vertices. Most terms in Eq. (4.4) involve flux variables which are known at cell faces. Interpolation is then used to compute the corresponding data at the cell vertices. To compute the curvature, Eq. (A.14) is used,

$$\kappa = \nabla_\Sigma \cdot \boldsymbol{\nu}. \tag{4.8}$$

For a surface of revolution, parametrized in the poloidal plane by its length s,

$$\begin{aligned} R &= \Phi(s), \\ Z &= \Psi(s), \end{aligned}$$

elaboration of Eq. (4.8) leads to a convenient analytical expression for the curvature. In a Cartesian coordinate system, the surface is described by

$$\mathbf{X}(s,\theta) = \begin{pmatrix} \Phi(s)\cos(\theta) \\ \Phi(s)\sin(\theta) \\ \Psi(s) \end{pmatrix}.$$

The unit vectors tangent to this surface are

$$\boldsymbol{\tau}_1 = \frac{\partial \mathbf{X}}{\partial s}\left\|\frac{\partial \mathbf{X}}{\partial s}\right\|^{-1} = \begin{pmatrix} \dot{\Phi}\cos(\theta) \\ \dot{\Phi}\sin(\theta) \\ \dot{\Psi} \end{pmatrix} \text{ and } \boldsymbol{\tau}_2 = \frac{\partial \mathbf{X}}{\partial \theta}\left\|\frac{\partial \mathbf{X}}{\partial \theta}\right\|^{-1} = \begin{pmatrix} -\sin(\theta) \\ \cos(\theta) \\ 0 \end{pmatrix},$$

while the unit normal is

$$\boldsymbol{\nu} = \pm\boldsymbol{\tau}_1 \times \boldsymbol{\tau}_2 = \mp \begin{pmatrix} \dot{\Psi}\cos(\theta) \\ \dot{\Psi}\sin(\theta) \\ -\dot{\Phi} \end{pmatrix}.$$

The sign of the normal must be chosen in order to have the outward pointing normal. Finally, elaborating Eq. (4.8) using expression (A.10), the curvature becomes

$$\kappa = \nabla_\Sigma \cdot \boldsymbol{\nu} = \mp \frac{\dot{\Psi} - \Phi\left(\dot{\Psi}\,\ddot{\Phi} - \ddot{\Psi}\,\dot{\Phi}\right)}{\Phi}. \tag{4.9}$$

4.2 Numerical Validation

This section is based on the research paper W. Dekeyser, D. Reiter, M. Baelmans, Optimal Shape Design for Divertors, IJCSE, in press [26]. In this paper, extensive numerical validation of the methods developed in this thesis is presented. The accuracy of the adjoint shape derivatives is evaluated, and the performance of the optimization algorithms is compared.

The results are based on a somewhat simplified edge plasma model, with corresponding changes in the adjoint model. The simplifications include neglecting the convective term in the neutral continuity equation, energy convection, as well as some of the momentum and energy source terms. In terms of the generalized convection diffusion model (3.35), the convection matrices are $C^\theta(\mathbf{q}) = \left(nu_\theta, mnu_\theta u_{||}, 0, 0\right)^T$ and $C^r(\mathbf{q}) = 0$, while the diffusion matrices simplify to $D^\theta(\mathbf{q}) = \mathrm{diag}\left(0, \eta^{\mathrm{i}}_\theta, D^{\mathrm{n}}_p, \kappa_\theta\right)$ and $D^r(\mathbf{q}) = \mathrm{diag}\left(D^{\mathrm{i}}, \eta^{\mathrm{i}}_r, D^{\mathrm{n}}_p, \kappa_r\right)$. The source terms in the equations are

$$S_{\mathrm{n}} = \begin{pmatrix} n_{\mathrm{e}}n_{\mathrm{n}}K_{\mathrm{i}} - n_{\mathrm{i}}n_{\mathrm{e}}K_{\mathrm{r}} \\ -mn_{\mathrm{i}}u_{||}(n_{\mathrm{e}}K_{\mathrm{r}} + n_{\mathrm{n}}K_{\mathrm{cx}}) \\ n_{\mathrm{i}}n_{\mathrm{e}}K_{\mathrm{r}} - n_{\mathrm{e}}n_{\mathrm{n}}K_{\mathrm{i}} \\ -E_{\mathrm{i}}n_{\mathrm{e}}n_{\mathrm{n}}K_{\mathrm{i}} \end{pmatrix}, \; S_z = \begin{pmatrix} 0 \\ 0 \\ 0 \\ -c_z n_{\mathrm{i}} n_{\mathrm{e}} L_z \end{pmatrix}, \; S_p = \begin{pmatrix} 0 \\ -\frac{b_\theta}{h_\theta}\frac{\partial p}{\partial \theta} \\ 0 \\ 0 \end{pmatrix}.$$

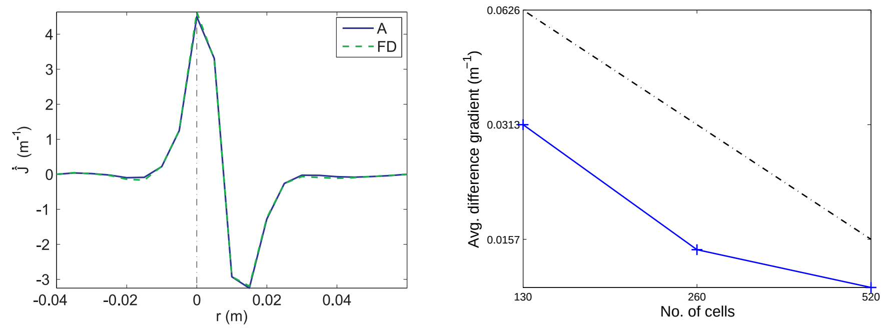
Using this simplified model leads to significant savings in computational time, while some dominant nonlinearities are still present. The model parameters are chosen to resemble a low density, low recycling (outboard half of) ITER problem, again for computational ease. The total length of the core boundary (EF) is 10 m, and the divertor legs are 1 m each, measured along AE and FD. The radial width of the simulated domain is 0.1 m, with radial decay lengths of 0.01 m applied as boundary conditions for plasma density and temperature at the wall and private flux boundaries. The targets are straight, with an angle of $\tan^{-1}(0.5) \approx 27^\circ$ between the poloidal magnetic field and the normal to the target surface. A constant pitch $b_\theta = 0.1$ is taken. Core density is set to $1 \cdot 10^{19}$ m^{-3}, and the input power is 50 MW. The aim is to keep the target heat flux as close to Q_{d} =0.1 MW m^{-2} as possible. Recycling coefficients of $R = 1$ are specified at target and wall boundaries, while pumping is modeled by a recycling and reflection coefficient of 0.9 at the private flux boundary. This corresponds to a thermal D_2 pumping speed of $\sim 2 \cdot 10^3$ m^3 s^{-1} — deliberately (very) high (e.g. typically a thermal D_2 pumping speed around 30 m^3 s^{-1} is assumed for ITER). An additional advantage of the model simplification is that it allows to use relatively coarse grids. For the numerical study, a reference grid with 130×20 cells was used.

4.2.1 Validation of Adjoint Shape Sensitivity Computation

The accuracy of the shape sensitivities computed using the continuous adjoint method can be evaluated by comparison with the derivative of the reduced cost functional obtained by a finite difference approach. In the latter method, a small perturbation ($\tau = 1 \cdot 10^{-7}$ m $\sim \sqrt{\epsilon_{\mathrm{mach}}}$) is applied to each design variable in turn, and the state equations are solved for each perturbed grid. The shape derivative is then

$$\dot{\hat{J}} \approx \frac{\hat{J}(\phi + \tau\delta\phi) - \hat{J}(\phi)}{\tau}. \tag{4.10}$$

Figure 4.2 (a) compares these two derivatives for the initial configuration of the test problem. The shape sensitivity along the target CD is shown. As expected,



(a) (b)

Figure 4.2: (a) Shape derivative of reduced cost functional computed with continuous adjoint method (A) and finite differences (FD) on a 130×20 grid, and (b) L1-norm of the difference between both gradients as the grid is refined two resp. four times in each coordinate direction. The dashed line in (b) represents theoretical first order convergence.

the sensitivity is the largest around the separatrix strike point (dashed line in the figure), as this is the region where the heat flux is concentrated. It is seen that the derivative computed with the adjoint method is accurate, even on the relatively coarse grid used for the test problem. In terms of computational cost, the difference is enormous (see Table 4.1), as the adjoint approach requires only the solution of the state and adjoint equations (approximately two flow simulations), while the finite difference method requires the solution of the state equations for every degree of freedom, i.e. 38 for the results shown here. As the perturbation is very small, the state equations converge in relatively few iterations (Table 4.1), yet this does not compensate for the large number of simulations.

As the grid is refined, the difference between the derivatives computed with finite differences and with the adjoint method should decrease due to a reduction of the discretization error. To illustrate this, the computation of the shape sensitivities with adjoint and finite difference methods is repeated on two additional grids which are successively refined by a factor of two in both poloidal and radial directions, i.e. with sizes 260×40 and 520×80. The results are summarized in Fig. 4.2 (b). In the figure, the L1-norm of the difference between the computed derivatives is plotted as a function of grid size. The figure shows that the error indeed reduces with the expected approximate first order accuracy of the hybrid schemes used in the code.

Table 4.1: Comparison of CPU times required for shape sensitivity computation.

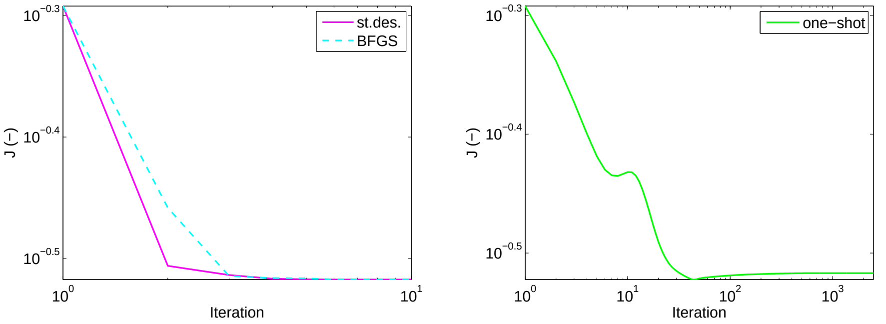
	Number of iterations	CPU time (s)	Relative CPU time (-)
Solve state equations	1400	490	1
Solve adjoint equations	1400	630	1.3
Compute gradient (FD)	38 × 600	7980	16.2
Compute gradient (A)	2 × 1400	1120	2.3
Steepest descent run	6	6180	12.6
BFGS run	7	4220	8.6
One-shot run	2500 (500)	2220 (930)	4.5 (1.9)

4.2.2 Comparison of Optimization Algorithms

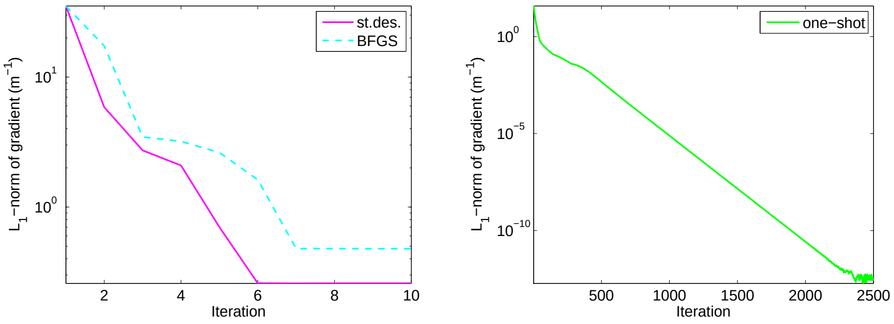
This section compares the steepest descent method and the BFGS method with the one-shot method in terms of performance. The former two methods are standard gradient-based optimization algorithms, while the latter attempts to directly solve the system of first order optimality conditions. The test problem is the same as used for the grid sensitivity analysis described above. By comparing the computational cost of the algorithms, the best performing optimization algorithm will be identified. This algorithm will then be used on more realistic design cases later on.

All optimization algorithms were started from the same initial configuration with straight targets. Time steps used for the state and adjoint equations are $1 \cdot 10^{-4}$ s. For the gradient-based algorithms, computational time was saved by performing a reduced number of iterations of the state equations during line searches. Similarly, sensitivity information required to evaluate the Wolfe conditions (for the BFGS method only) is based on partially converged adjoint equations. For the one-shot method, 2500 iterations of state and adjoint equations were performed. Also the grid was updated 2500 times, but for the cylindrical shell model this requires negligible computational effort. The update of the design equation (see Eq. (2.36)) was done by using the smoothed steepest descent direction from Eq. (2.37) and a constant step length $\alpha = 1 \cdot 10^{-4}$ m^2. The latter step length was determined experimentally as the largest stable step size. This step size corresponds to approximately 10% of the poloidal length of cells at the target.

Figure 4.3 shows how the value of the cost functional and the L1-norm of the shape derivative are reduced during the optimization runs. As the one-shot method does not use any information on the value of the (discretized) cost functional J, it is able to solve the design equation up to machine precision.



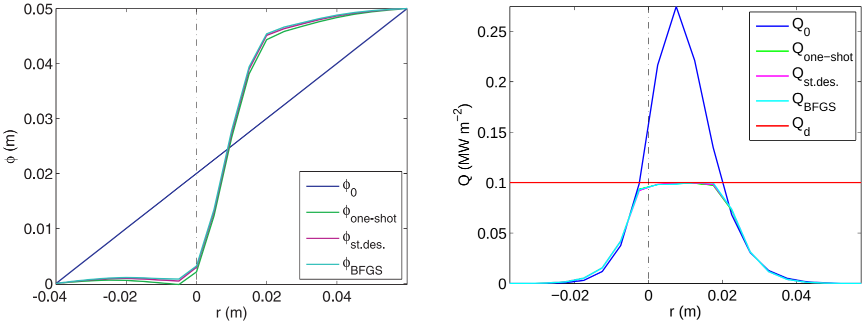
(a) Cost functional (b) Cost functional



(c) Shape derivative (d) Shape derivative

Figure 4.3: Cost functional and L1-norm of the shape derivative for the different optimization methods.

Thus, the L1-norm of the gradient is reduced by several orders of magnitude. Note that the value of the cost functional does not decrease monotonically. The steepest descent and BFGS methods both use line searches on the value of the cost functional in order to determine appropriate step lengths. After a certain number of iterations (6 resp. 7 for the tests performed), these methods stall. This can be attributed to discretization effects. First of all, when closing in on the optimum shape, the shape derivative computed with the continuous adjoint method is no longer a descent direction for the discretized optimization problem. Furthermore, the solution to the discrete problem becomes less and less accurate on the strongly distorted grids near the optimum, leading to further differences between the continuous problem and its discrete approximation. Note that this effect would not occur when using the discrete adjoint approach, where the adjoint equations are derived based on the discretized state equations and cost functional, and thus provide the exact derivative of the discretized problem. Depending on the point of view, the stalling effect could be regarded as an advantage, as it gives a clear indication when discretization errors are starting to dominate in the numerical code. Thus, it can be argued that at this point, the accuracy of the discrete simulations has been reached. Figure 4.4 shows that indeed all three methods have found approximately the same solution. In terms of discrete cost functional value, the BFGS solution is slightly better than the solution found with the steepest descent method, which in turn is somewhat better than the one-shot solution. Comparing L1-norms of the shape derivative, the ranking is exactly opposite.



(a) Optimized target profiles

(b) Optimized energy fluxes

Figure 4.4: Optimized target profiles and corresponding heat fluxes obtained with the different optimization algorithms. Subscripts ‘0’ refer to the initial, straight target.

Comparing the relative CPU times given in Table 4.1, it is seen that the one-shot method finds a solution to the optimization problem in less than 5 times the time needed for a single state simulation. This is significantly faster than the gradient-based optimization algorithms. It is interesting to see that the steepest descent algorithm requires approximately the same number of optimization iterations as the BFGS method. This is probably due to the gradient smoothing, which mimics the behavior of the Hessian. Still, the BFGS method is considerably faster because better initial step lengths are produced, and thus less time is spent on the line search. In fact, five out of seven times, the first step length was accepted, and thus no time was lost on the line search at all.

The comparison of CPU times is unfair in the sense that the CPU times given for steepest descent and BFGS methods include only the computational time required to reduce the L1-norm of the gradient by two orders of magnitude and then converge the solution of the state equations to machine precision, while for the adjoint method all equations are converged to machine precision. If the one-shot loop is interrupted after 500 iterations, and then the state equations are further converged to machine precision, the same accuracy is reached as in the gradient-based methods, but at an even lower cost. These numbers are given in parentheses in Table 4.1.

4.3 Target Shape Optimization in Simplified SOL Geometry

In this section, results of divertor target optimization using the edge plasma model of Chapter 3 are presented. Section 4.3.1 is devoted to a detailed analysis of the shape sensitivities. Next, the one-shot approach is used to optimize divertor targets for uniform power load for a number of problems in Section 4.3.2.

4.3.1 Shape Sensitivities

The shape sensitivities are analyzed based on a JET-sized model problem similar to the one shown in Figure 4.1. Core density is set to $2 \cdot 10^{19}$ m^{-3}, and the input power is 3 MW. The aim is to keep the target heat flux as close to Q_d =0.5 MW m^{-2} as possible. The domain is up-down symmetric, with the total length of the core boundary (EF) 9 m, and the lengths of the divertor legs each 0.5 m, measured along AE and FD. The radial width of the simulated domain is 0.1 m, with radial decay lengths of 0.03 m for plasma density and 0.3 m for the temperature applied as boundary conditions at the wall and private flux boundaries. A straight target with an angle $\alpha = \tan^{-1}(0.5) \approx 27°$ between the poloidal magnetic field and the normal to the target surface is taken. Combined with the pitch $b_\theta = 0.075$, this means an angle of $\sin^{-1}(b_\theta \cos\alpha) \approx 4°$ between the magnetic field itself and the target. Recycling coefficients $R = 0.9$ are specified at target and wall boundaries, while pumping is modeled by an absorption coefficient of $\alpha_\mathrm{p} = 0.02$ at the private flux boundary. This corresponds to a thermal D_2 pumping speed of ~ 130 m^3 s^{-1}, representative of the JET vacuum pumping system [18, 122]. Transport parameters are summarized in Table 4.2 In order to resolve the gradients developing at the target, a grid of 280×80 cells in the poloidal and radial directions is used.

First, as a validation step, the adjoint shape sensitivity is compared to the one obtained through finite differencing. The importance of the nine-point stencil is

Table 4.2: Parameters of the JET-sized model problem.

Parameter	Value	Parameter	Value
D^{i}	1.0 $\mathrm{m}^2\ \mathrm{s}^{-1}$	χ^{n}	1.4
ν^{i}	0.2 $\mathrm{m}^2\ \mathrm{s}^{-1}$	c_z	0.01
χ^{i}	1.0 $\mathrm{m}^2\ \mathrm{s}^{-1}$	E_{i}	25 eV
χ^{e}	1.0 $\mathrm{m}^2\ \mathrm{s}^{-1}$		

illustrated by computing sensitivities for the same problem while neglecting the non-orthogonality of the cells. Then, the critical issue of gradient smoothing is discussed.

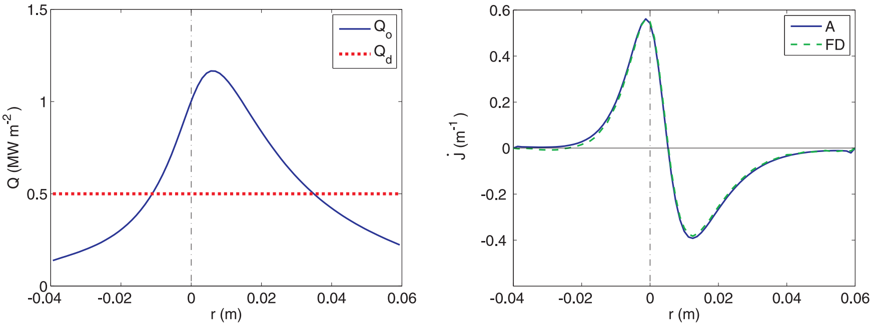
Accuracy of the adjoint shape sensitivity

Figure 4.5 compares the adjoint and finite difference sensitivities for the test problem. It is confirmed that also in this case the derivative computed with the adjoint method is accurate. Due to the finer grid, the benefit in terms of computational cost is even higher now. The adjoint approach requires only the solution of the state and adjoint equations (two flow simulations of approximately 10.000 iterations with time step 10^{-5} s each), while the finite difference method requires the solution of the state equations for every degree of freedom, now 2×79 simulations. The simulation on the perturbed geometry converges in approximately 2.000 iterations. Thus, the computational cost is roughly a factor of 15 smaller with the adjoint approach in this case.

During optimization, the target geometry becomes more complex compared to the initial straight target configuration. Therefore, also the grid becomes more irregular. Usually its quality decreases due to the increased distortion. Figure 4.6 compares the grids at the initial configuration and at a configuration close to an optimum. In order to achieve good resolution of the gradients at the sheath boundary, the grids are strongly refined in this area. In the nearly optimized configuration, this leads to large cell distortion. In Figure 4.7 the adjoint and finite difference shape sensitivities on this last grid are compared. In general, qualitative agreement is still retrieved, but there are some larger quantitative differences now. This is due to a combination of the grid distortion and the proximity to the optimal solution, where the continuous adjoint shape sensitivity may start to deviate from the discrete sensitivity. This example clearly shows that sufficiently fine grids are needed for optimization.

Importance of the nine-point stencil

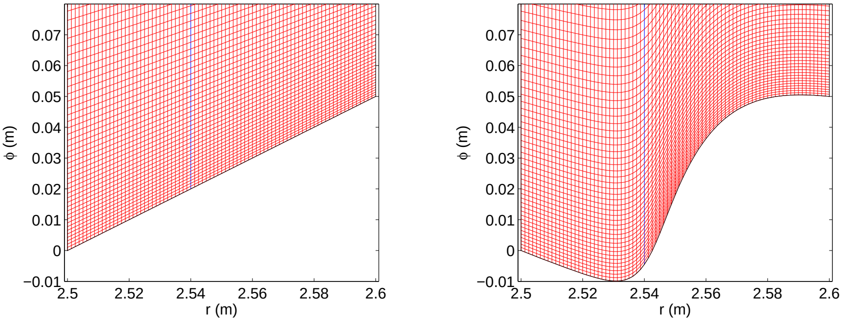
In order to assess the importance of the nine-point stencil, the JET-like model problem described above is repeated with identical parameters, only this time leaving out all nine-point corrections for non-orthogonal grids in the discretization scheme. Both adjoint and finite difference sensitivities are evaluated for the resulting five-point scheme. The energy flux to the target



(a)

(b)

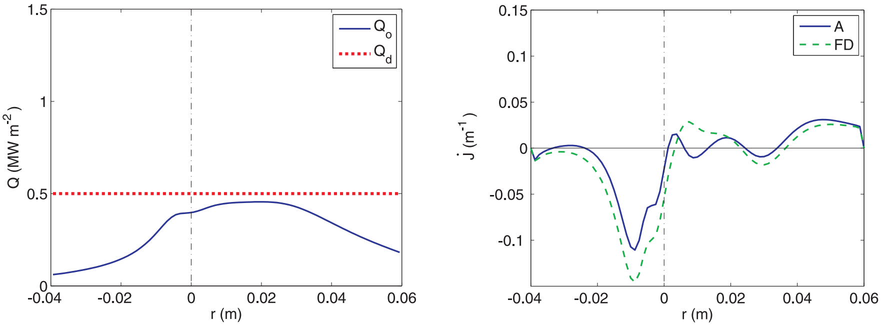
Figure 4.5: (a) Total and desired energy flux to the target, and (b) shape derivative of reduced cost functional computed with continuous adjoint method (A) and finite differences (FD) on a 280×80 grid at initial target geometry.



(a)

(b)

Figure 4.6: (a) Initial grid, configuration with straight target. (b) Intermediate configuration, strongly distorted grid.

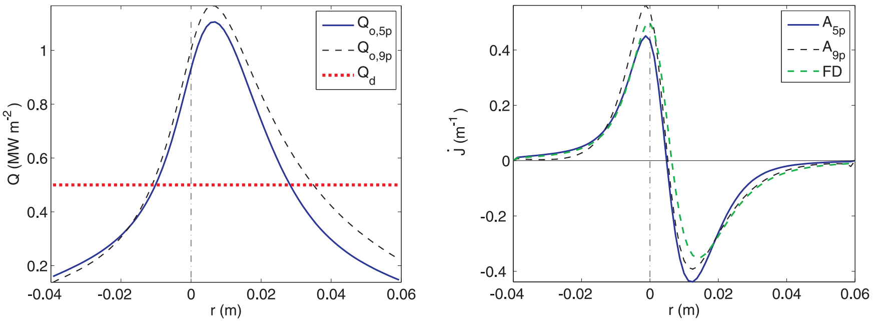


(a) (b)

Figure 4.7: (a) Total and desired energy flux to the target, and (b) shape derivative of reduced cost functional computed with continuous adjoint method (A) and finite differences (FD) on a 280×80 grid at the intermediate configuration of Fig. 4.6 (b).

and the shape sensitivities are shown in Fig. 4.8. For ease of comparison, the corresponding energy flux and shape sensitivities from Fig. 4.5, computed with the full nine-point scheme, are also added in the figure. It is seen that the simulated energy flux by itself differs significantly between the simulations. This is mainly because radial energy transport to the targets is not present in the five-point scheme, while it can locally constitute up to 20% of the target energy flux in the nine-point case. An important part of this radial energy flow is conducted by neutrals. The five-point adjoint and finite difference shape sensitivities agree qualitatively, but there is a significant quantitative difference, Fig. 4.8 (b), especially in comparison to the correspondence achieved with the nine-point stencil (see Fig. 4.5). In an edge plasma code as B2, the five-point approximation neglecting grid distortion is usually applied. However, since the neutrals are treated kinetically by a Monte Carlo code (EIRENE), their transport is not affected by this approximation. The dominant parallel transport of the plasma component may make this problem less pronounced. For fluid neutral models, on the other hand, this effect should be taken into consideration when implementing the adjoint methods. For numerical consistency, hybrid continuous-discrete adjoint methods as discussed at the end of Chapter 2 may be an option. However, such methods would hide but not eliminate the numerical errors.

Before moving on, the profiles of some plasma properties obtained with nine-point and five-point schemes are further investigated, Fig. 4.9. In the figure, densities and temperatures are compared at the midplane and at the targets.



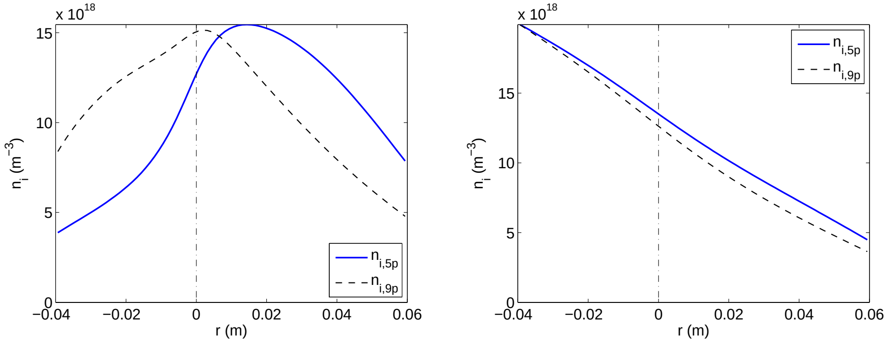
(a)

(b)

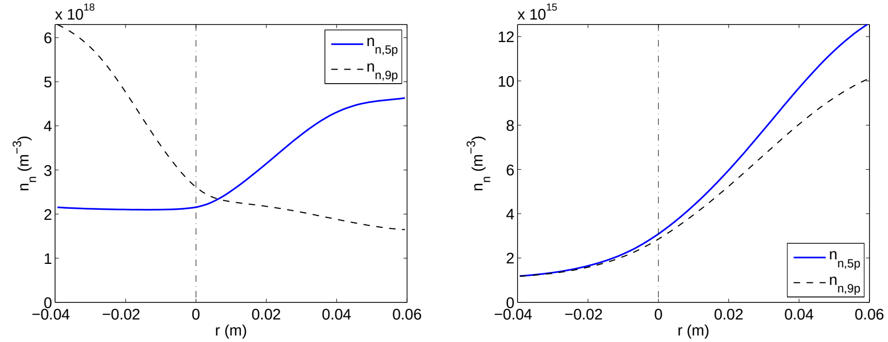
Figure 4.8: (a) Total and desired energy flux to the target, and (b) shape derivative of reduced cost functional computed with continuous adjoint method (A) and finite differences (FD) on a 280×80 grid, using a five-point rather than nine-point discretization scheme. For comparison, also the results obtained with the nine-point stencil are shown in dashed lines (cf. Fig. 4.5).

Although the results at the midplane agree quite well between the five and nine-point cases, the predicted target conditions are qualitatively very different. It is especially interesting to see for example the peak in neutral densities at the opposite ends of the target. The five-point scheme tends to compress neutrals in the obtuse angle at the far SOL, while the nine-point scheme leads to compression in the acute angle of the private flux region. Qualitatively, compression in the acute corner seems to agree better with kinetic neutral results, see for example [60]. However, detailed model benchmarking is required to confirm this. This type of differences in predictions of divertor conditions is probably a more demanding reason to consider full nine-point discretizations in edge codes as B2, particularly in case of fluid neutrals.

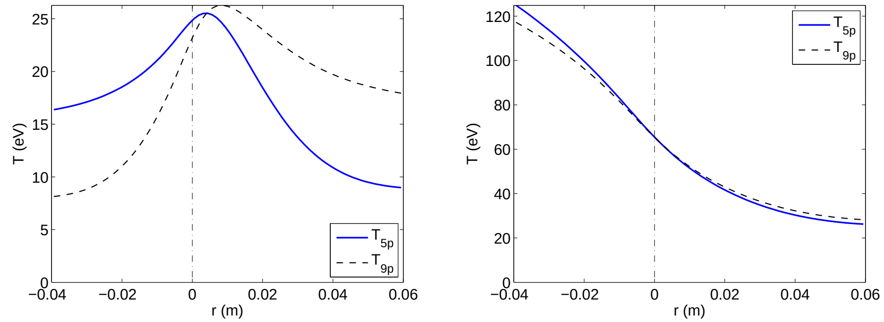
To close this section, it should be noted that the modeling issue of the nine-point solver could naturally be identified by using the continuous adjoint method. In this particular case, the large discrepancy between the sensitivities computed with finite differences and with the continuous adjoint method was due to hidden approximations in the discrete implementation, and not purely due to discretization effects.



(a) Ion density, target (b) Ion density, midplane

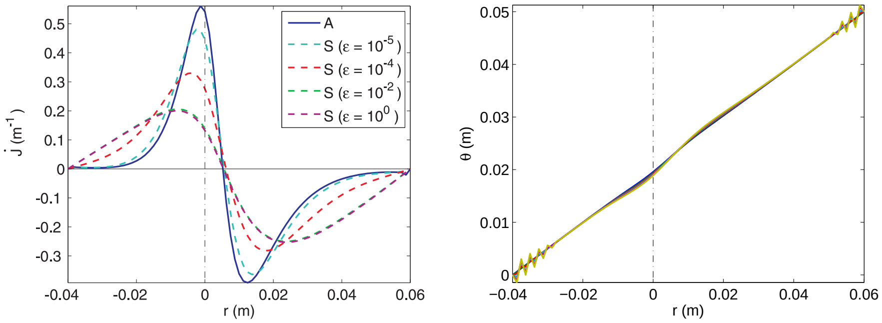


(c) Neutral density, target (d) Neutral density, midplane



(e) Temperature, target (f) Temperature, midplane

Figure 4.9: A comparison of target and midplane profiles computed with five-point and nine-point stencil.



(a) (b)

Figure 4.10: (a) Adjoint gradient before (A) and after (S) smoothing. (b) Target oscillations appearing in one-shot run without gradient smoothing.

Gradient smoothing

In Section 2.4.2, it has been pointed out that smoothing the gradient is important to preserve the regularity of the domain. Fig. 4.10 (a) shows the gradients of the JET-like model problem before and after smoothing, for different values of the smoothing parameter ϵ in Eq. (2.37). After smoothing, the gradient is rescaled by requiring

$$\int_{\Sigma} \left|\tilde{\mathcal{G}}\right| \, \mathrm{d}\sigma = \int_{\Sigma} \left|\mathcal{G}\right| \, \mathrm{d}\sigma.$$

For $\epsilon = 10^{-5}$, the smoothed gradient resembles the one before smoothing quite well. By increasing ϵ, more and more of the ‘fine structure’ of the gradient is removed. Still, all smoothed gradients remain descent directions.

To illustrate the effect of *not* smoothing the gradient, a one-shot optimization run is started from this initial configuration, where the design variable is repeatedly updated with the non smoothed gradient. After only a few iterations, oscillations appear on the targets, Fig. 4.10 (b), and the algorithm soon crashes. Using the smoothed gradient, this is avoided.

A relatively high value of $\epsilon = 1 \cdot 10^{-2}$ has been used for all optimization runs presented below. The motivation to use a high value of the smoothing parameter is to avoid getting trapped in local optima in the early stages of the simulation. Since the optimization algorithm is driven by the gradient, the obtained solution will always be some local solution. However, by initially allowing only very smooth deformations, the ‘macroscopic’ behavior is captured first, and only close to the optimal solution details in the shapes appear.

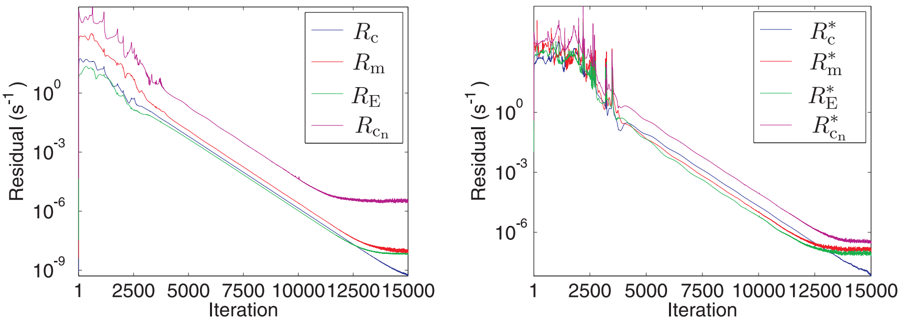
4.3.2 Optimized Target Geometries

In this section, the results of divertor target shape optimization in ‘slab’ geometry are presented. Since the one-shot method emerged as the most efficient optimization method in Section 4.2.2, it is also used for the problem studied here. Results of divertor optimization in slab geometry with the reduced model presented in Section 4.2 have been published in W. Dekeyser, D. Reiter, M. Baelmans, Divertor Design through Shape Optimization, Contributions to Plasma Physics 52 (2012), 544-549 [28], and with a somewhat more elaborate model in W. Dekeyser, D. Reiter, M. Baelmans, Adapting computational optimization concepts from aeronautics to nuclear fusion reactor design, EPJ Web of Conferences 33 (2012), 03009 [27]. Both papers deal with high recycling ITER-like problems. Since the method has meanwhile been extended to the full model of Chapter 3, this thesis will focus only on these last results for brevity.

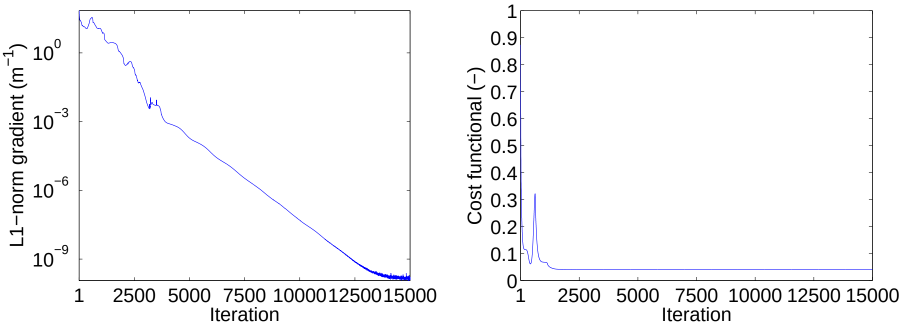
The sample problem used in this section is the JET-like case described in Section 4.3.1. All model parameters and the initial configuration with straight targets are described there. For the present cases, the recycling coefficient is increased to $R = 1$, so particle exhaust only takes place at the pumps at the private flux boundary. The aim is to achieve a uniform load at the targets. Two levels of this uniform load are considered. In a first, ‘easy’ problem the desired uniform energy flux is $Q_{\mathrm{d,h}} = 0.5$ MW m^{-2}. Then, a more challenging case with a lower desired flux $Q_{\mathrm{d,l}} = 0.25$ MW m^{-2} is studied.

Before diving into the results, it is remarked again that grid quality is essential for accuracy of the sensitivity computation, as well as for the stable solution of the adjoint problem. Furthermore, the importance of using the nine-point stencil in the correction equations emerged in these test cases. Initially, it was tried to solve these optimization problems using a five-point stencil scheme in the correction equations (but still using the full nine-point discretization in the computation of residuals). Solver performance then gradually declined as the grids deformed more and more during optimization. Finally, the update scheme became unstable and convergence was not achieved. Especially the (adjoint) neutrals suffer from the five-point correction treatment, because their transport is isotropic while grids are field aligned. By performing updates using the nine-point stencil, full performance is regained.

The time step used to relax state and adjoint equations during the one-shot optimization run is the same as during the solution of the state or adjoint equations alone (10^{-5} s in this case). The residuals of state and adjoint equations, as well as the L1-norm of the shape sensitivity and the value of the cost functional are shown in Fig. 4.11. This figure corresponds to the $Q_{\mathrm{d,l}}$ case. It is seen that after some initial, erratic behavior, the residuals nicely converge



(a) Residuals state equations (b) Residuals adjoint equations



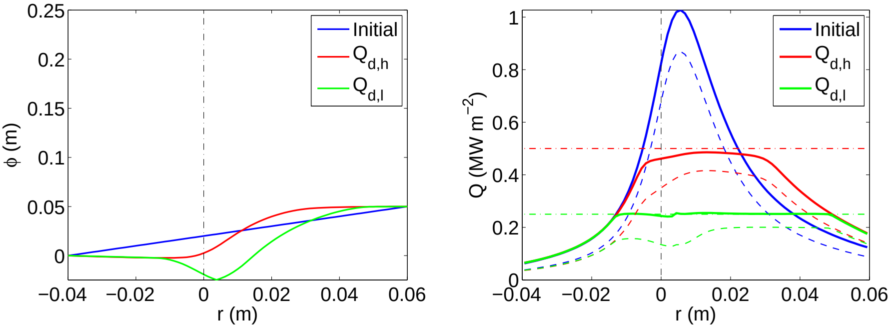
(c) L1-norm shape sensitivity (d) Cost functional

Figure 4.11: Evolution of the residuals, shape sensitivity and cost functional during optimization ($Q_{\mathrm{d,l}}$ case).

to machine precision. The same is true for the shape sensitivity. In the one-shot scheme, the cost functional does not necessarily decrease monotonically.

The target profiles and energy fluxes after optimization are shown in Fig. 4.12. Compared to the initial configuration, some important differences can be noticed. First of all, the optimizer tends to incline the targets locally in order to increase the plasma-wetted area and reduce the peak energy flux density. Furthermore, also the local plasma conditions are modified in order to reduce the peak energy flux density and increase the spreading of the power load.

In the case with $Q_{\mathrm{d,h}}$ as desired flux, the increased surface area is the main reason for the peak energy load reduction. This is illustrated in Fig. 4.13. On the left figure, the quantity $\mathbf{Q}_\mathrm{o} \cdot \mathbf{e}_\theta = (\delta_\mathrm{sh} T + E_\mathrm{p})\, n u_\theta$ is plotted. This can be

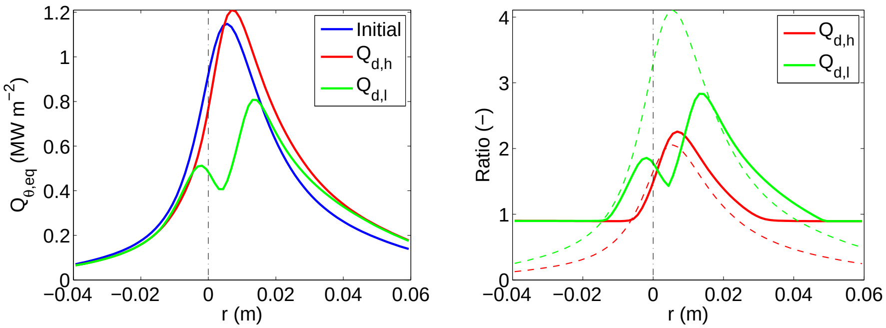


(a) Target profiles (b) Energy fluxes

Figure 4.12: Comparison of initial and optimized target profiles and the resulting energy fluxes. The solid lines in (b) are the total energy flux Q_o, while the dashed lines represent Q_t. The difference between the two is the surface recombination energy, Q_sr.

interpreted as an equivalent poloidal flux to the target (equivalent, because in practice the energy flux to the target contains poloidal and radial contributions). This quantity allows to compare how much energy is deposited locally. In the right figure, the ratio of initial and desired energy flux densities is compared to the ratio of optimized and initial surface areas. For this high Q_d case, the increased target area even allows to locally *increase* the deposited power compared to the initial configuration. This increase is due to the increased temperature along the right part of the target, see Fig. 4.14 (c). The modified temperature profile even allows to deposit a larger total power at the target and increase the spreading, without exceeding the 0.5 MW m^{-2} limit. The global balances of ion particles and energy for the initial and optimized configurations are given in the first two columns of Table 4.3. Clearly, the optimization did not have a large impact on the global balances. Similarly, Fig. 4.14 confirms that upstream quantities hardly changed.

For the $Q_\mathrm{d,l}$ case, the picture is quite different. Now, the increase in local target area only accounts for approximately half of the peak energy load reduction. At the location where the initial energy flux density is highest, the optimized target area is even lower than in the $Q_\mathrm{d,h}$ case. Now, the energy load is brought down by playing on the neutral particles. A 'V-shaped' structure appears in the target, which acts as a bowl to trap the neutrals locally at the point of highest energy flux. This has several effects. Since the neutrals are trapped, their density strongly rises. This is clearly visible in the neutral density profile at the target, Fig. 4.14 (e), and also in the contour plot Fig. 4.15 (b). Comparison



(a) (b)

Figure 4.13: The left figure compares the power deposited along the target for the initial and optimized configurations. On the right, the ratio of initial and desired energy flux densities is shown in dashed lines, and the ratio of optimized and initial surface areas in solid lines.

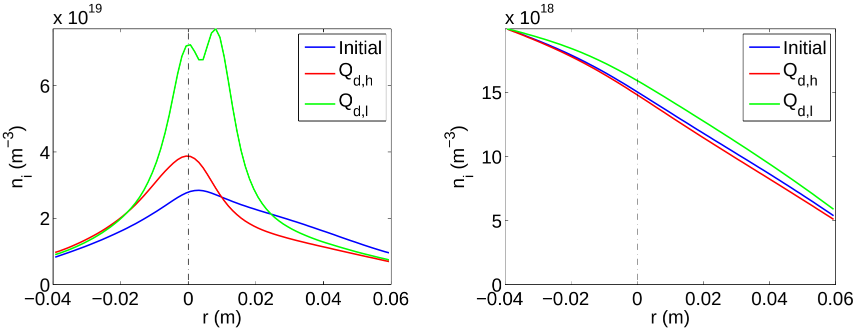
with Fig. 4.15 (a) shows that in the initial configuration the neutrals were trapped in the private flux area, where they interact less with the plasma. The trapping of neutrals in the V-shaped area leads to a much more intense recycling process. This is reflected in the increased energy loss due to plasma-neutral interactions, Fig. 4.15 (d). Indeed, at every ionization event, the ionization energy has to be provided by the plasma. Furthermore, the increased neutral density leads to a correspondingly high plasma density, Fig. 4.14 (a), which in turn leads to an increased energy loss due to impurity radiation, Fig. 4.15 (f). Finally, the increased energy loss from the flux tubes combined with a strong momentum sink due to charge-exchange interactions leads to partial detachment of the plasma in these flux tubes. The detachment is already indicated by the inverted temperature profile at the target, Fig. 4.14 (c). Indeed, the temperature is now lowest close to the separatrix strike point. A more convincing signature of detachment is found in the comparison of the total pressure upstream to the one at the target, Fig. 4.16. In an attached plasma, these total pressures should be approximately equal in the SOL. This is indeed the case for the initial configuration, and also for the $Q_{\mathrm{d,h}}$ case. For the $Q_{\mathrm{d,l}}$ case, on the other hand, there is a clear drop in total pressure in the flux tubes of the V-shaped configuration, which indicates detachment. In summary, the third column in Table 4.3 gives the global ion particle and energy balances for this optimized configuration. The more intense plasma-neutral interaction is reflected in the increased ion flux to the targets, as well as in the higher volumetric source. The total energy deposited at the targets is now reduced by

Table 4.3: Comparison of ion particle and energy balances at initial and optimized configurations.

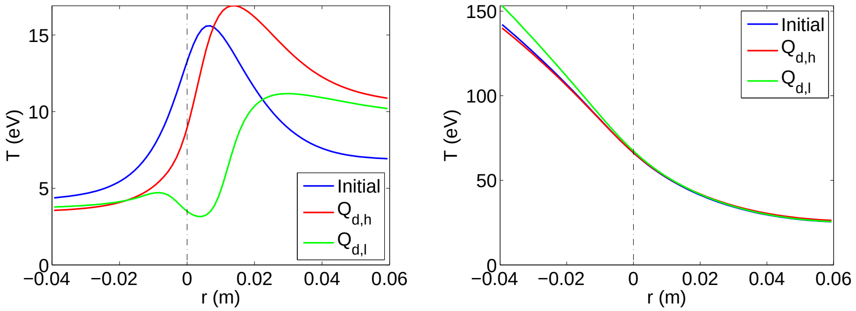
	Initial	Optimized, 0.5 MW m^{-2}	Optimized, 0.25 MW m^{-2}
Particle balance (10^{20} s^{-1})			
Flux from core	122.1	130.8	84.8
Flux to targets	2 × 661.2	2 × 672.3	2 × 782.1
Flux to outer wall	347.7	309.9	345.6
Flux to PF	76.9	83.3	63.8
Volumetric source	1624.9	1607.0	1888.8
Energy balance (MW)			
Flux from core	3.00	3.00	3.00
Flux to targets	2 × 0.52	2 × 0.57	2 × 0.40
Flux to outer wall	0.41	0.42	0.43
Flux to PF	0.08	0.07	0.06
Neutral interactions (sink)	0.33	0.34	0.45
Impurity radiation (sink)	0.70	0.64	0.83
Other volumetric sources	-0.43	-0.38	-0.44

20%. This is compensated by increased energy losses due to impurity radiation and ionization. In part, the latter energy is also radiated. Although the radiated energy is usually distributed more isotropically across the first wall and targets, a significant part of this additional radiation originates quite close to the target, and may be redeposited rather locally. Therefore, the obtained configuration may not be optimal anymore if the radiation energy becomes dominant. This problem will be studied in Chapter 8.

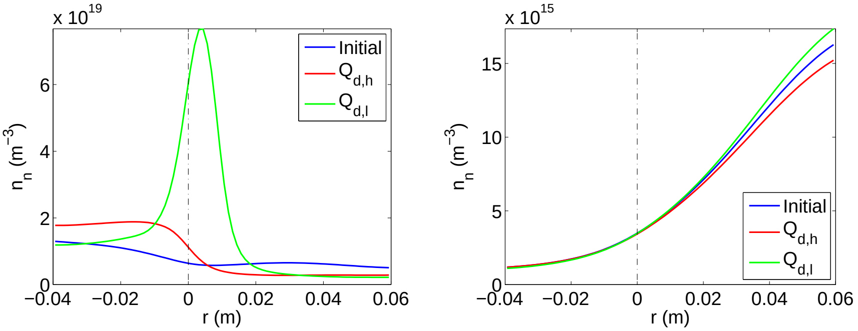
Although the slab model used in this chapter is quite a large geometric simplification of reality, already some interesting design features have been retrieved with these test problems. For example, strongly inclined targets and V-shaped configurations have been retained in the design of ITER [75]. It has been recognized that highly radiating and (partially) detached regimes will be essential to bring down the heat load in large-scale devices. The optimization algorithms developed in this thesis in a sense *prove* that these features are optimal with respect to obtaining a low, uniform energy load. Furthermore, by fully taking the interaction of the plasma with the divertor geometry into account, a flat heat load is obtained over a large part of the target. It would be very difficult to achieve similar results without the aid of detailed shape sensitivity analysis.



(a) Ion density (b) Ion density

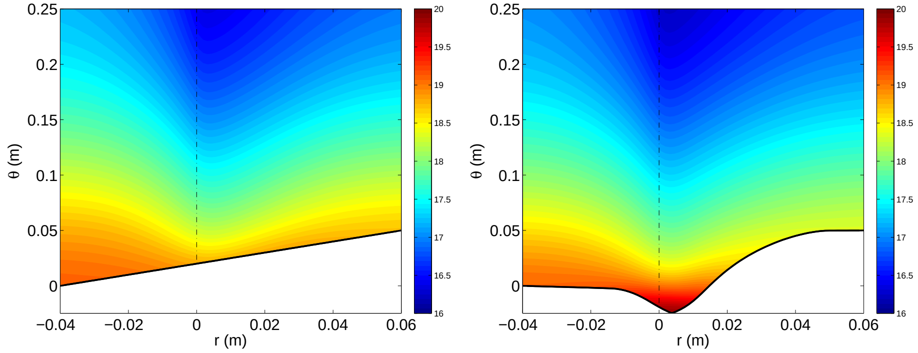


(c) Temperature (d) Temperature



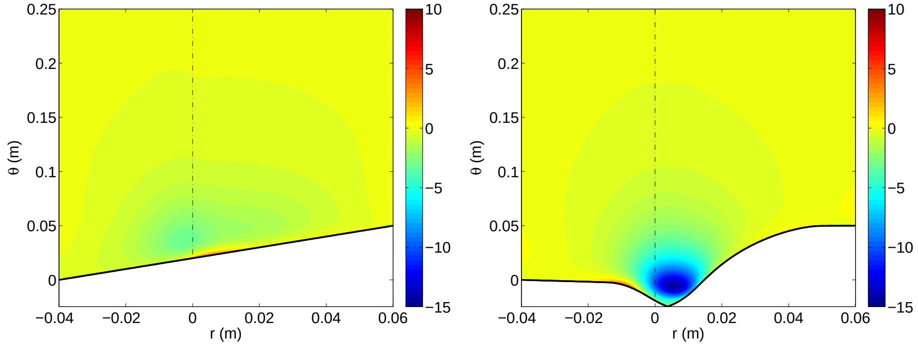
(e) Neutral density (f) Neutral density

Figure 4.14: Comparison of profiles at the targets (figures on the left) and at the midplane (figures on the right) for the initial and optimized configurations.



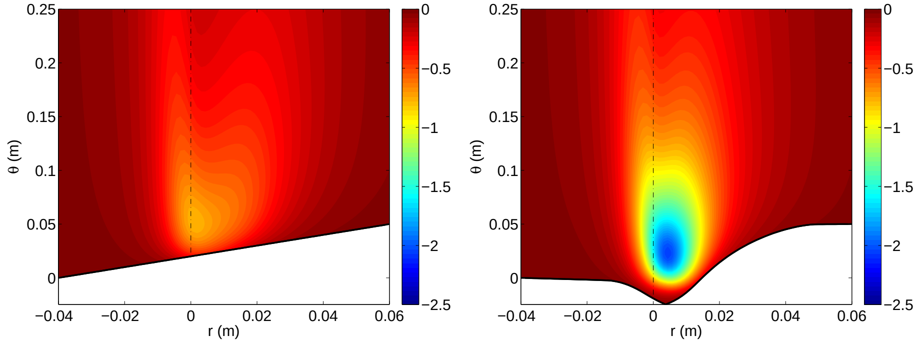
(a) Neutral density ($\log_{10}(n_{\mathrm{n}})$)

(b) Neutral density ($\log_{10}(n_{\mathrm{n}})$)



(c) Energy loss ionization (MW m^{-3})

(d) Energy loss ionization (MW m^{-3})



(e) Impurity radiation (MW m^{-3})

(f) Impurity radiation (MW m^{-3})

Figure 4.15: Comparison of the energy balance of initial (figures on the left) and optimized (figures on the right, $Q_{\mathrm{d,l}}$) target designs.

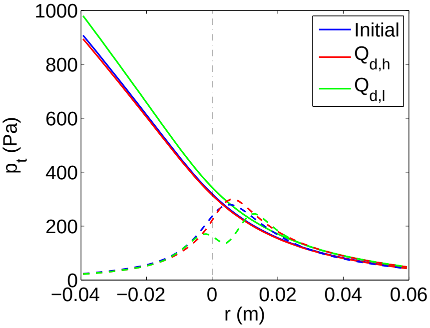


Figure 4.16: Comparison of total pressure at upstream (solid lines) and target (dashed lines) locations for initial and optimized cases. Note that for $r < 0$, the upstream and target locations do not correspond to the same flux tube. The high upstream pressure is in the core region, while the low target pressure is in the private flux area.

4.4 Conclusion

Extensive numerical validation and of the developed algorithms is presented in this chapter. First, the shape sensitivities computed with the velocity method are compared to their finite difference counterparts. It is confirmed that the sensitivities are accurate, and converge to the same solution if the grid is refined. The importance of the correct nine-point discretization of the fluxes on non-orthogonal grids is shown. In the first place, the nine-point scheme appears to be essential for the correct simulation of the fluid neutrals. Furthermore, it is illustrated that also the shape sensitivity suffers from an incorrect discretization.

Next, the one-shot method is compared to two gradient-based optimization algorithms: the steepest descent and the BFGS method. It is shown that even if the gradient-based methods use only partially converged state and adjoint solutions, they are still outperformed by the one-shot approach. Therefore, the latter method will be the method of choice in the rest of the thesis.

Finally, the one-shot algorithm is applied to a JET-like model problem in simplified edge plasma geometry. The targets are optimized for two levels of the uniform power load. If the peak load is only moderately reduced, the optimized target configuration can handle a larger integrated flux compared to the initial configuration. If the peak load is reduced further, the plasma partially detaches. In this case, less power reaches the target. This is mainly compensated by increased losses through neutral ionization and impurity radiation.

Chapter 5

Divertor Shape Optimization in Realistic SOL Geometry

In the previous chapter, the shape optimization methods are applied to model problems in simplified slab geometry. This allowed for extensive numerical validation and performance comparison of different optimization algorithms. In the present chapter, the shape optimization algorithms are extended to realistic tokamak geometry. The set-up of the optimization problem is discussed in Section 5.1. Due to the general formulation of the optimization problem in curvilinear coordinates in Chapter 3, the state and adjoint equations remain unchanged. However, the extension to realistic edge geometry requires adaptation of the grid generation algorithm. This will be treated in Section 5.2. Then, optimization results will be discussed in Section 5.3.

5.1 Problem Formulation

As in Chapter 4, divertor target design for uniform power load is considered. The cost functional to be minimized is the same as in Eq. (4.1),

$$J(\Omega, \mathbf{q}) = \frac{1}{2} \int_{\mathrm{t}} \left(Q_{\mathrm{o}} - Q_{\mathrm{d}}\right)^2 \,\mathrm{d}\sigma, \tag{5.1}$$

where $Q_{\mathrm{o}} = Q_{\mathrm{t}} + Q_{\mathrm{sr}}$ has contributions from heat convection and conduction as well as from surface recombination, see Eq. (4.2), and Q_{d} is the desired uniform load. The integral is taken along the inner and outer targets, including part of the baffles. This part of the divertor is shown with a thick blue line in Fig. 5.1.

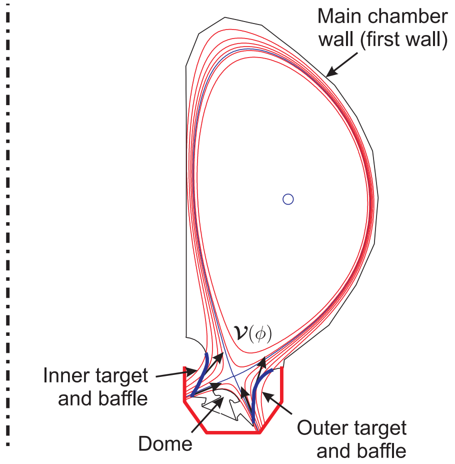


Figure 5.1: Poloidal cross-section of ITER (ITER F57 equilibrium [75]), indicating the design parameterization used for the targets.

Since wide grids [6] are not used in this thesis, the simulated domain is restricted to the last field lines running from target to target and tangent to either the main chamber wall or the dome. Only the corresponding parts of the targets and baffles (indicated with a thick blue line in Fig. 5.1) are allowed to move. The targets are parametrized by a function ϕ which determines their displacement along the magnetic field lines, see Fig. 5.1. By definition, $\phi = 0$ in the initial configuration, with a positive value of ϕ indicating a displacement of the target towards the plasma side. As before, the end points of the target are kept fixed. Additionally, the design space may be restricted by limiting the movement of the targets to stay inside the vacuum vessel. This type of constraint is illustrated by the thick red line in Fig. 5.1, but will not be an active constraint in the optimization results presented later in this chapter.

5.2 Grid Generation

Grid generation for edge plasma computations is not trivial. The strongly anisotropic transport in the edge plasma poses a challenge for numerical simulations. The highest numerical accuracy can be obtained by using nearly quadrilateral grid cells, with two *radial* cell faces aligned with the magnetic flux surfaces, and the other two *poloidal* faces as orthogonal to the field as possible. The large gradients in the radial direction, especially near the separatrix, require a good radial grid resolution, while cells can be considerably longer in the poloidal direction, leading to high aspect ratio grid cells. The high aspect ratio is also important for numerical accuracy, since it makes the integrated fluxes across poloidal and radial cell faces of the same order of magnitude. In the divertor area, on the other hand, the grid has to be adapted in order to match the divertor geometry (baffles, targets,...). Typically highly distorted cells result, since target plates are often strongly tilted with respect to the magnetic field. Also, considerable grid refinement in the poloidal direction is needed in order to resolve the strong gradients developing towards the sheath and for the proper resolution of the fluid neutrals.

Grid generation proceeds in several steps. First, the magnetic field $\mathbf{B} = \mathbf{B}_\theta + \mathbf{B}_\phi$ has to be specified, where $\mathbf{B}_\theta$ and $\mathbf{B}_\phi$ are the poloidal and toroidal field, respectively. For toroidally symmetric systems, the field can always be written as [33]

$$\mathbf{B} = I(\Psi)\nabla\phi + \frac{1}{2\pi}\nabla\phi \times \nabla\Psi, \tag{5.2}$$

where $\nabla\phi = \frac{1}{R}\mathbf{e}_\phi$.

To see this, consider the poloidal disk flux Ψ [33],

$$\Psi = -\int_{S_\mathrm{d}} \mathbf{B}\cdot \mathrm{d}\mathbf{S} = -\int_{S_\mathrm{d}} (\nabla \times \mathbf{A})\cdot \mathrm{d}\mathbf{S} = -\oint_{\Gamma_\mathrm{d}} \mathbf{A}\cdot \mathrm{d}\mathbf{l} = -2\pi R A_\phi.$$

A_ϕ is the toroidal component of the magnetic vector potential $\mathbf{A}$. The disk S_d, with boundary Γ_d is everywhere tangent to the magnetic flux surface, Fig. 5.2. Due to the toroidal symmetry, A_ϕ is constant along the integration path. By definition, Ψ is constant on a flux surface (i.e. Ψ is the same for any disk tangent to a particular flux surface), and can thus be used as a flux surface label. The poloidal disk flux Ψ is completely determined by the poloidal magnetic field, so the latter can be derived from it by

$$\mathbf{B}_\theta = \nabla \times \mathbf{A}_\phi = \nabla \times (RA_\phi \nabla\phi) = \frac{1}{2\pi}\nabla\phi \times \nabla\Psi. \tag{5.3}$$

Thus, $\Psi/(2\pi)$ is a stream function for the poloidal magnetic field.

The toroidal field $\mathbf{B}_\phi$ is directed along $\nabla\phi$. Therefore, $\mathbf{B}_\phi = I\nabla\phi$ and $B_\phi = I/R$. In ideal MHD, $I = I(\Psi)$ is also a flux function, and is related to the total poloidal current flowing through the disk in Fig. 5.2 [33]. Assuming the (poloidal) currents in the edge plasma are negligible, I is approximately constant in the edge and can be written as $I = B_{\phi,0}R_0$ with $B_{\phi,0}$ the toroidal field at a reference radius R_0. Thus, the toroidal field in the plasma edge is assumed to have a $1/R$ dependence.

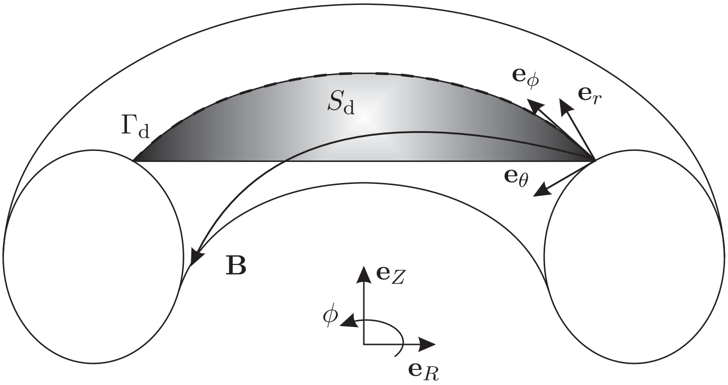


Figure 5.2: Disk tangent to the flux surface.

Once the magnetic field is known, the X-point and the magnetic axis can be determined as points where $\mathbf{B}_\theta = 0$, i.e. $\nabla\Psi = 0$. The X-point will determine the separatrix. Then, the geometry of the vessel is added to the picture. It is assumed that the geometry is specified as a (set of) piecewise linear polygon(s). In order to have a good spatial resolution at the targets, grid cells are spaced at smoothly varying intervals of Ψ along one of the targets. The corresponding flux surfaces are then traced to the other target by a contouring algorithm. One of these flux surfaces must be the separatrix. The flux surfaces inside the separatrix are again determined by smoothly varying values of Ψ between X-point and magnetic axis, and are traced by the contouring algorithm. In a divertor configuration, the magnetic field expands significantly in the vicinity of the X-point. When the flux surface spacing at the target and inside the separatrix is determined, it is important to keep in mind that the grid must still be sufficiently fine at the X-point. In practice, this means that flux surfaces at the target and at upstream positions will be bunched quite closely around the separatrix.

Once all flux surfaces have been determined, the ‘radial’ coordinate lines are computed. Therefore, first a smooth distribution of cells along the separatrix is specified. By definition, the $\nabla\Psi$ field is everywhere locally orthogonal to the flux surfaces, so its streamlines can be used as radial coordinate lines inside the separatrix and also in the SOL far from the targets. This ensures that cells are (nearly) orthogonal to the magnetic field, and increases numerical accuracy. Furthermore, since flux surfaces are strongly bunched at the separatrix this near orthogonality is essential in order to avoid cell twisting in areas of strong magnetic field curvature, see Ref. [11]. When moving closer to the targets, grid lines are gradually deformed to match the target surface. Since steep gradients

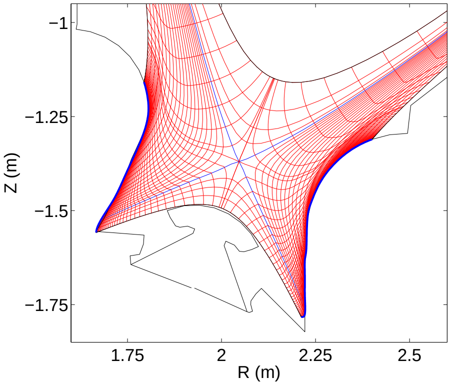


Figure 5.3: Zoom on divertor area of a grid for a (scaled) ITER F57 equilibrium. In order to improve visibility, only every second radial line and every tenth flux surface is shown. The actual grid has 220×240 cells in the poloidal and radial directions, respectively.

develop towards the sheath, poloidal cell sizes are strongly refined in this area.

Figure 5.3 shows a typical grid for an ITER configuration, with a zoom of the X-point area. For such strongly inclined targets, high quality grids with both a good target and a good X-point resolution can only be achieved by using a large number of cells. The grid in the figure has 220 cells in the poloidal and 240 cells in the radial direction. It is also seen that due to the strong inclination, grid cells become very small at the baffles. Since all fluxes are correctly discretized with a nine-point scheme by the edge plasma code, there is no need for an optimization step to make all angles as close to orthogonal as possible as in for example Carre [81]. However, a smoothing algorithm to avoid the extreme bunching of coordinate lines at the baffles and the small cells resulting from (almost) orthogonal radial lines around the X-point should be considered for further improvement of the grid quality.

Grid morphing

For numerical accuracy and the reliable computation of sensitivities, high quality grids are required. Inevitably, this means grid generation becomes an expensive task. Depending on the resolution, creating a single grid may take up to several minutes, mainly due to the requirement of near-orthogonality of the poloidal faces upstream and inside the separatrix. Indeed, this requires tracing of the coordinate lines perpendicular to the flux surfaces and finding the intersections with the latter. If orthogonality is not required, as in the divertor area, grid vertices can simply be distributed along the surfaces, a much cheaper procedure.

The grid generation algorithm has to be executed at every iteration of the optimization loop, see Fig. 3.7 (b) in Section 3.5.3. In the one-shot framework — which appeared as the most efficient candidate for the solution of the optimization problem in Chapter 4 — the number of optimization iterations can be very large, and thus the grid generator becomes a bottleneck. In order to have an efficient overall algorithm, the CPU time spent on the grid generator should only be a small fraction of the CPU time required for performing one relaxation step of state and adjoint equations. Fortunately, the majority of the cells — and in particular all cells which are orthogonal to the magnetic field — are situated far from the divertor area, and thus don't have to be deformed as the divertor geometry changes during optimization. They can be computed in a preprocessing step. Only the cells close to the divertor targets should move to match the changing target geometry, but this can be achieved very efficiently by simple stretching of the cells along the magnetic flux surfaces. The cost of this morphing procedure is negligible.

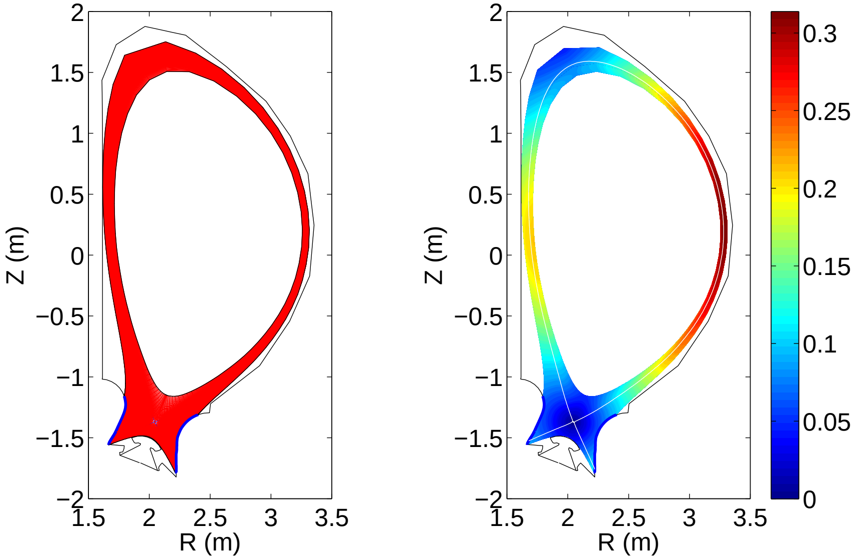
5.3 Optimization in Realistic Toroidal Geometry

In this section, optimization in realistic SOL geometry is explored. The test problem is described first, followed by a density scan to qualitatively assess the model. Then, the accuracy of the shape sensitivities is evaluated by comparison with the finite difference method. Finally, results of divertor target optimization are presented.

5.3.1 Model Problem

The ITER F57 equilibrium [75] is used as the initial configuration, but the geometry is scaled by 40% in both axial and radial directions to obtain a JET-sized problem (Fig. 5.4). The target configuration is optimized for a case with 3 MW core input power and fixed core density of $3 \cdot 10^{19}$ m^{-3}. The recycling coefficient is uniformly set to $R = 1$ at all boundaries, and a thermal D_2 pumping speed of ~ 130 m^3 s^{-1} is assumed. Pumping is assumed to take place along the entire private flux boundary. An overview of all relevant model parameters and boundary conditions is given in Table 5.1.

The choice of geometric scaling factors and model parameters in this case study is mainly driven by computational cost. While the geometric complexity is the



(a) Geometry (b) Magnetic field pitch

Figure 5.4: Initial configuration: scaled ITER F57 equilibrium [75].

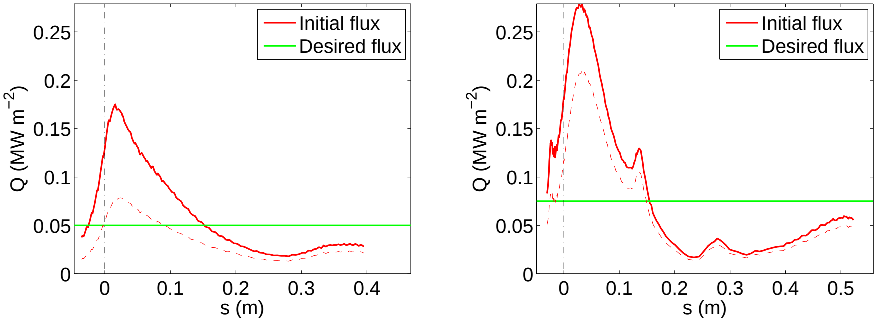
same as in larger-scale devices, the power is representative of an L-mode JET case and not of ITER. This setup allows for much faster convergence of the state and adjoint equations, and is therefore more suited for the validation of the design methodology.

The initial total target heat loads (convection, conduction and surface

Table 5.1: Model parameters and boundary conditions.

Model parameters	Value	Boundary conditions	Value
D^{i}	$1.0\ \mathrm{m}^2\ \mathrm{s}^{-1}$	Q_{c}	3.00 MW
ν^{i}	$0.2\ \mathrm{m}^2\ \mathrm{s}^{-1}$	n_{c}	$3 \cdot 10^{19}\ \mathrm{m}^{-3}$
χ^{i}	$1.0\ \mathrm{m}^2\ \mathrm{s}^{-1}$	δ_{sh}	4.8
χ^{e}	$1.0\ \mathrm{m}^2\ \mathrm{s}^{-1}$	λ_n	0.03 m
χ^{n}	1.4	λ_T	1.00 m
c_z	0.01	R	1.00
E_{i}	25 eV	α_{p}	0.05

recombination) can be seen in Fig. 5.5. The simulated loads show some small oscillations. This is due to a combination of the roughness of the underlying magnetic field data, the piece wise linear representation of the target surface and the large grid distortion. The aim of optimization will be to reduce the peak load by roughly a factor of four to 0.05 MW m^{-2} for the inner target and 0.075 MW m^{-2} for the outer target.



(a) Inner target (b) Outer target

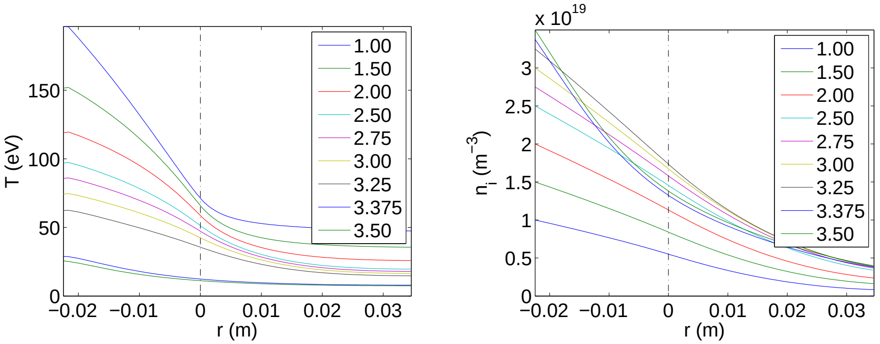
Figure 5.5: Target energy loads in initial configuration. The horizontal axis is the distance along the target, with $s = 0$ at the separatrix strike point, $s > 0$ in the SOL and $s < 0$ in the private flux. The solid lines represent the total load (convection, conduction and surface recombination), while the dashed lines contain only convection and conduction.

For a good resolution of the targets and X-point, a grid with 220×240 cells is used. A zoom of this grid can be seen in Fig. 5.3. The time step for simulation of the state equations is 10^{-5} s. The problem converges in approximately 7.500 iterations. It is remarked that the use of the nine-point scheme in the correction equations was essential in order to achieve convergence with the state and adjoint solvers on this highly distorted grid.

5.3.2 Density Scan

In order to get a feeling for the state of the plasma, a density scan is performed for the initial configuration. All model parameters are kept fixed, except for the core density n_c.

Figure 5.6 shows the ion density and temperature profiles at the outer midplane. As the core density rises at fixed power, the temperature decreases.



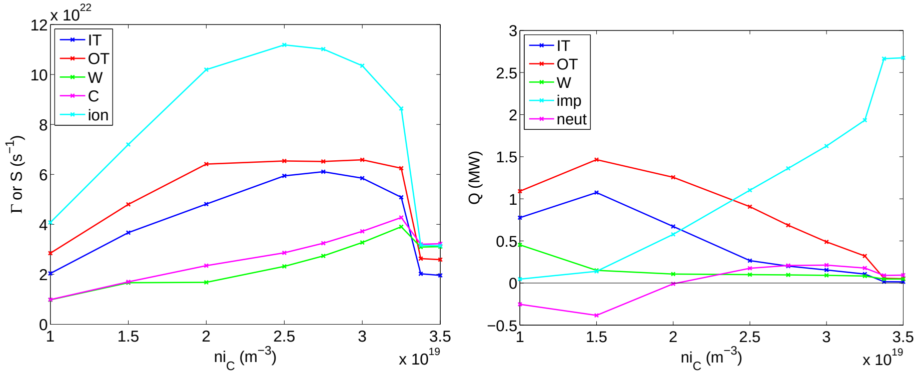
(a) Temperature (b) Ion density

Figure 5.6: Outer midplane profiles for different values of the core density n_c. The legend gives the value of n_c in 10^{19} m^{-3}.

The integrated ion and energy fluxes to the different boundaries are examined in Figure 5.7. The ion fluxes from the core and to the wall scale quite linearly with core density. The ion fluxes to both targets initially also increase. At a certain point (at $n_\mathrm{c} \approx 2.75 \cdot 10^{19}$ m^{-3} for the inner target and $n_\mathrm{c} \approx 3 \cdot 10^{19}$ m^{-3} for the outer target), the target fluxes 'roll over' and start to decrease again. After this roll-over, detachment of the inner and then the outer targets gradually sets in. Finally, at $n_\mathrm{c} \approx 3.3 \cdot 10^{19}$ m^{-3}, there is a sudden drop in all ion fluxes.

Except at very low densities, the energy fluxes to the targets decrease with increasing core density. This decrease is compensated by a gradual increase in impurity radiation, which in turn is a result of increasing divertor density and a reduction of the divertor temperature to the level where carbon strongly radiates, see Fig. 3.2 in Section 3.1. The radiating zone gradually moves from the targets to the X-point. At very high density, the temperature in the closed field line region has become very low, and a radiating mantle is created inside the separatrix, mainly on the high field side. Almost all the core power is radiated before it enters the SOL. The inner target almost completely detaches, while the outer target only detaches close to the separatrix strike point. The formation of the radiating mantle explains the sharp drop in ion and energy fluxes. Furthermore, since the SOL temperature is now very low the neutrals recycling from the main chamber wall penetrate inside the separatrix and are ionized there. This leads to the steeper ion density profiles, Fig. 5.6.

At first sight, the net effect of neutral interactions on the energy balance is counter intuitive. At low density, neutral ionization leads to an internal energy sink. Although ionization and therefore the corresponding energy sink becomes



(a) Ion fluxes and sources

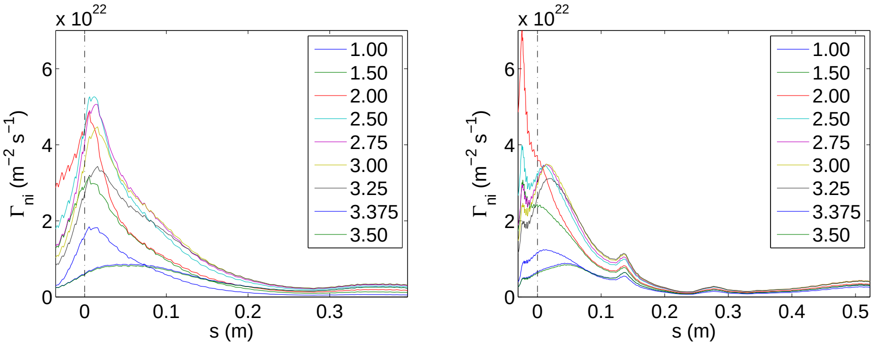
(b) Energy fluxes sources

Figure 5.7: Integrated ion and energy fluxes as a function of core density. IT: flux to inner target, OT: flux to outer target, W: flux to wall, C: flux from core, ion: total (net) volumetric ion source, imp: total energy sink due to impurity radiation, neut: total energy source due to neutral interactions. Fluxes to the private flux boundary are relatively small, and not shown in the figures.

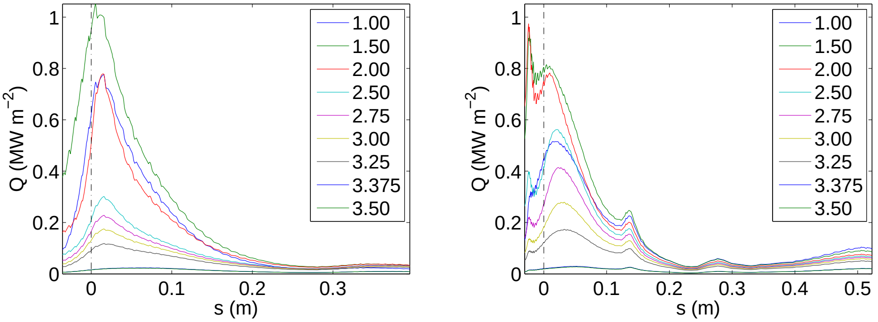
stronger with increasing density, this contribution is off set by collisional heating. As a result, at higher recycling the neutrals tend to heat rather than cool the plasma.

Figure 5.8 shows the detailed ion and total energy fluxes to inner and outer targets. Due to the geometric details of the outer target, there is some very local trapping of neutrals in the furthest part of the V-shaped corner resulting in the somewhat surprising peaks in the ion fluxes at low recycling (low core density) conditions. At higher density, the neutral presence becomes more pronounced in a larger part of the divertor, and this local effect disappears. However, this local effect is responsible for the fast initial rise of the integrated ion flux to the outer target and the rather flat integrated flux to the outer target for $n_{\mathrm{c}} \approx 2 - 3 \cdot 10^{19}\ \mathrm{m}^{-3}$. Furthermore, it tends to hide the fact that the SOL flux tubes at the outer target roll over at higher density than at the inner target. This can be seen more clearly in the density roll-over, Figure 5.9. In fact, in terms of peak density the outer target does not roll over before the radiating mantle is formed.

The profiles of temperature, ion density and neutral density and temperature are given in Fig. 5.10. At low core density, the temperature is highest near the separatrix strike point. As the core ion density increases, the temperatures decrease correspondingly. Finally, when the targets start to detach a temperature inversion is seen: low strike point temperatures and high temperatures in the



(a) Ion flux, inner target (b) Ion flux, outer target



(c) Energy flux, inner target (d) Energy flux, outer target

Figure 5.8: Target fluxes as a function of core density n_c. The legend gives the value of n_c in 10^{19} m^{-3}.

far SOL. Neutral profiles show the expected behavior. The neutrals are dragged along by the ion flow and compressed in the deep V-shaped corners. Their density increases with core density. At full detachment, the temperature is low in the entire divertor area and they start to fill this entire volume, albeit at somewhat reduced density.

Figure 5.7 shows that at the lowest core density, there is a sudden increase in the energy flux to the outer wall. The origin of this high heat flux can be understood by considering in detail the decay length conditions imposed at this boundary. To this end, the energy flux to the wall,

$$Q_r = \frac{5}{2} \sum_a \Gamma_r^a T - \kappa_r \frac{1}{h_r} \frac{\partial T}{\partial r},$$

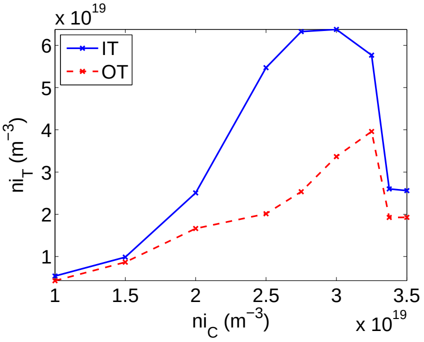
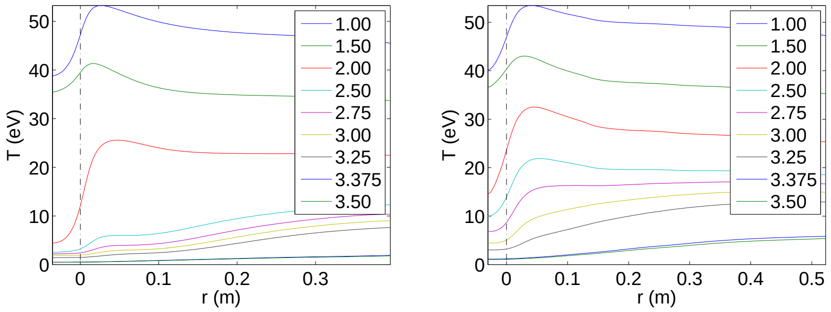


Figure 5.9: Maximum density at inner and outer targets as a function of core density.

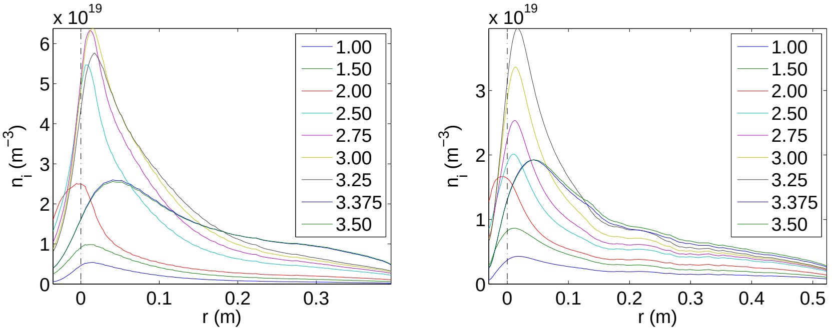
is elaborated with boundary conditions (3.28):

$$Q_r = \left(\frac{5}{2} \left(1 + Z_\mathrm{i} - R\right) + \frac{\left(\chi^\mathrm{i} + Z_\mathrm{i}\chi^\mathrm{e}\right)}{D^\mathrm{i}} \frac{\lambda_n}{\lambda_T} + \frac{\chi^\mathrm{n} D^\mathrm{n}_p}{D^\mathrm{i}} \frac{p_\mathrm{n}}{n} \frac{\lambda_n}{\lambda_T} \right) \Gamma^\mathrm{i}_r T.$$

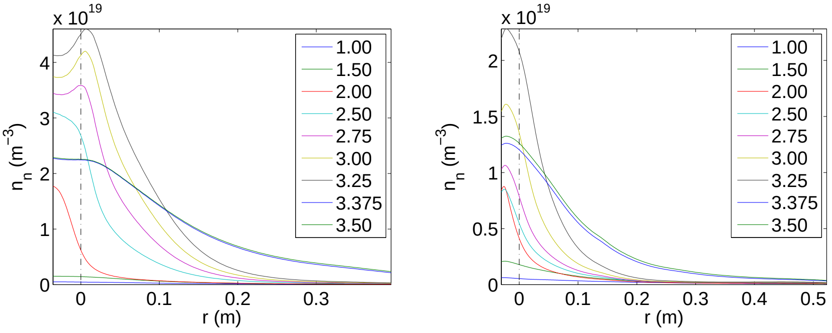
The term in parentheses is can be interpreted as an *effective heat transmission coefficient*. It has three contributions: one due to heat convection, a second due to ion and electron heat conduction, and a third term due to neutral heat conduction. For typical parameter values, the first contribution is of order unity and the second scales as λ_n/λ_T. On the other hand, the neutral conduction term can be of order 10^2 or higher in low density cases. Indeed, the neutral pressure diffusion coefficient scales as n^{-1}. Thus, at low n, the last term completely dominates the heat transferred to the wall, in a regime where the pressure diffusion model is strictly speaking not valid, because it was derived assuming dominant ion-neutral collisions (large n). Clearly, this is an unwanted artifact of the model. It explains why the linear scaling of target flux with upstream density in the low recycling regime is not retrieved. Indeed, the artificially high D^n_p transports all energy radially towards the outer wall rather than along the field to the divertor targets if the plasma density is too low. The unwanted effect is reduced by increasing the temperature decay length. Alternative options include imposing a gradient zero condition for the temperature, reducing the neutral thermal conductivity χ^n [21], or flux limiting the neutral pressure diffusion coefficient [21]. Since the model is mainly intended to be used in the high density, high recycling regimes relevant to fusion reactors, only the increased temperature decay length is used here.



(a) Temperature, inner target (b) Temperature, outer target

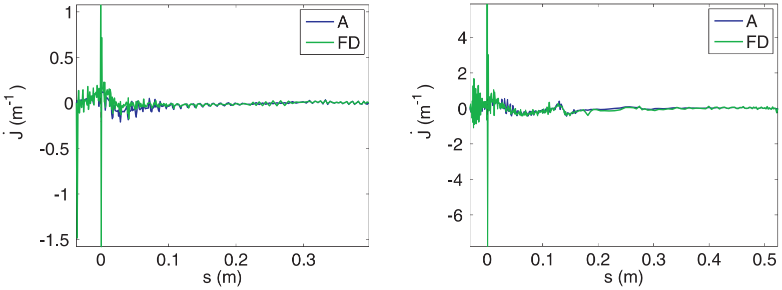


(c) Ion density, inner target (d) Ion density, outer target



(e) Neutral density, inner target (f) Neutral density, outer target

Figure 5.10: Profiles of temperature, ion density and neutral density at the targets as a function of core density n_c. The legend gives the value of n_c in 10^{19} m^{-3}.



(a) Inner target (b) Outer target

Figure 5.11: Shape derivatives computed with continuous adjoint method (A) and finite differences (FD) at both targets. The 'spikes' in the gradients are due to the piecewise linear representation of the targets.

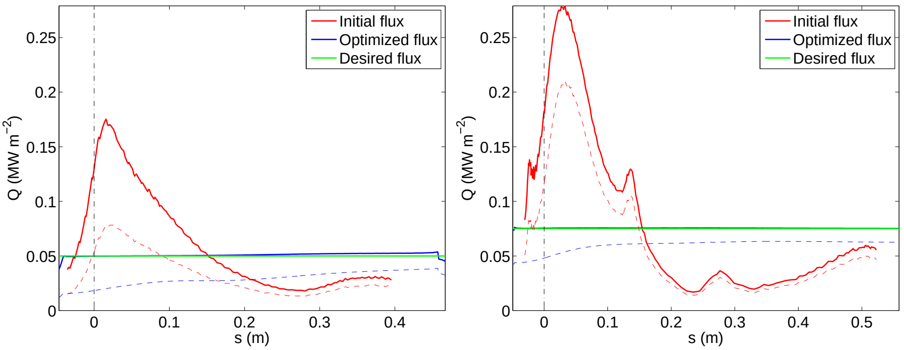
5.3.3 Validation of Shape Sensitivity Computation

The accuracy of the adjoint shape sensitivity is assessed by comparing with the finite difference approach. The result is shown in Fig. 5.11. As can be seen in the figure, the qualitative agreement is good, but there are still some quantitative differences. Although a grid sensitivity analysis was not performed for this case, it is expected that these differences are mainly due discretization effects. Clearly, the grid is still relatively coarse for use with the continuous adjoint approach. Furthermore, the large grid distortion required to resolve the geometry magnifies the discretization errors. The spikes in the shape sensitivity are due to the piece wise linear representation of the vessel geometry. At the intersection between two pieces of vessel, the curvature is not defined (tends to infinity). This effect appears to be more pronounced in case of finite differences. Since the optimization is performed with a smoothed gradient, these spikes do not influence the robustness of the optimization algorithm.

In terms of computational time, use of the adjoint method is very beneficial here. Since the grid has $N_r = 240$ cells in the radial direction, there are $2\,(N_r + 1) = 482$ design variables to characterize the inner and outer targets. For each perturbed design variable, 2.500 iterations were required to converge the state equations (compared to 7.500 for a full simulation). Therefore, evaluating the shape sensitivity with finite differences has an equivalent cost of ~ 160 state simulations, whereas adjoint sensitivity analysis requires only the solution of state and adjoint equations. Thus, a reduction of computational cost with a factor of approximately 80 is achieved.

5.3.4 Divertor Geometry Optimization

In this section, the results of the target shape optimization are presented. Looking back at the density scan, it is seen that the core density of $3 \cdot 10^{19}$ m^{-3} corresponds to an inner target which is already partially detached, while the outer target is still fully attached.



(a) Inner target (b) Outer target

Figure 5.12: Total target energy loads before and after optimization. Solid lines represent the total energy load, dashed lines the sum of conducted and convected energy.

The optimized total heat fluxes to the targets are shown in Figure 5.12. The desired flat loads have been matched almost perfectly along the entire target. The peak load has thus been reduced by a factor of four, and all target material is fully exploited. The energy balance in Table 5.2 reveals that the total load of the targets is 6% higher in the optimized case, while impurity radiation has been somewhat reduced. The table also shows that the ion fluxes to the targets have been reduced. This is interesting because it would reduce sputtering.

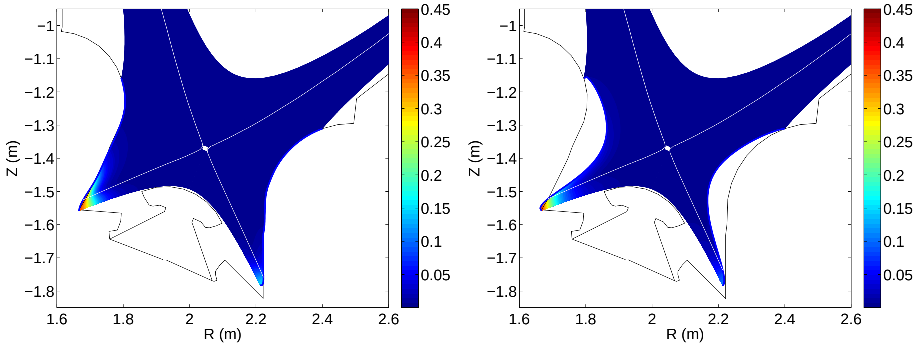
The optimized configuration is shown in Figure 5.13. The divertor has been squeezed inwards, and two rather long divertor legs are produced. In this particular case, however, the motivation for the long legs is not in the increased neutral interaction. Indeed, Figures 5.13 (a) and (b) show that in fact the neutral densities in the optimized configuration are lower around the separatrix strike points. The optimized divertor is in a state somewhat further from detachment. Figures (c) and (d) show that the energy sink due to neutral interactions is reduced (less ionization). Comparing the impurity radiation in figures (e) and (f) leads to similar conclusions. Profiles of some state variables along inner and outer targets are compared in Fig. 5.14. In the figures, the

Table 5.2: Comparison of particle and energy balances at initial and optimized configurations .

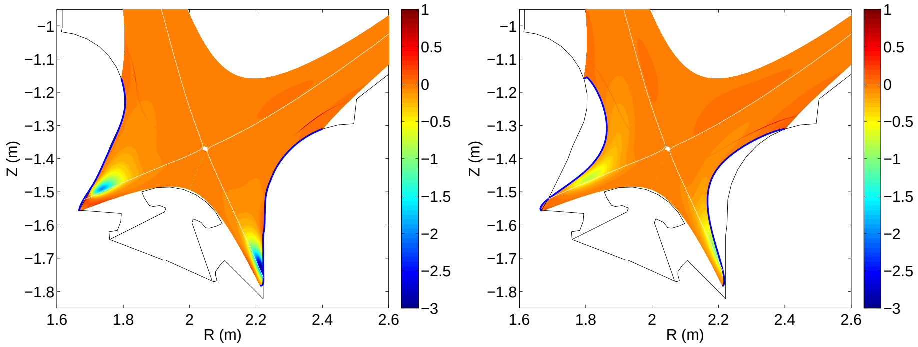
	Initial	Optimized
Particle balance (10^{20} s^{-1})		
Flux from core	372.0	350.6
Flux to inner target	585.0	582.7
Flux to outer target	658.3	608.0
Flux to outer wall	119.2	122.5
Flux to PF	44.6	37.3
Volumetric source	1035.1	999.9
Energy balance (MW)		
Flux from core	3.00	3.00
Flux to inner target	0.15	0.17
Flux to outer target	0.49	0.51
Flux to outer wall	0.09	0.09
Flux to PF	0.08	0.07
Neutral interactions (sink)	0.21	0.18
Impurity radiation (sink)	1.63	1.59
Other volumetric sources	-0.35	-0.39

profiles of the optimized geometry are ‘stretched back’ to the coordinates of the initial configuration, so this coordinate now acts as a ‘radial’ coordinate. It is seen that the temperatures near the strike points are higher in the optimized configuration, while the densities are a bit lower there.

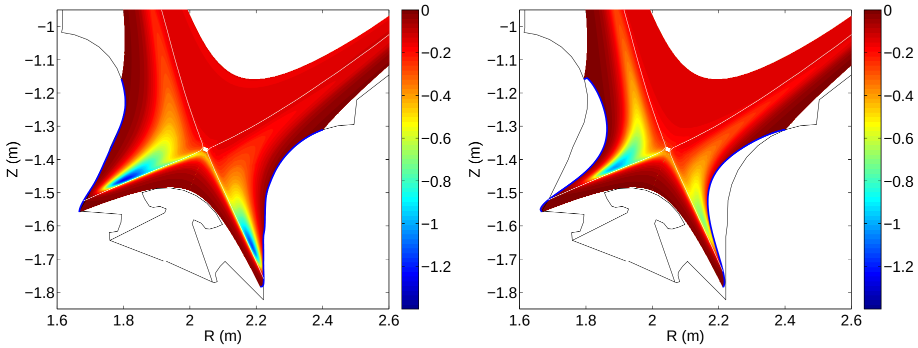
Upstream profiles are not significantly modified by the change in divertor design, as confirmed in Fig. 5.15. This is an interesting result in its own. Apparently divertor conditions can be modified quite significantly by playing with the geometry without influencing the main equilibrium.



(a) Neutral density (10^{20} m^{-3}) (b) Neutral density (10^{20} m^{-3})

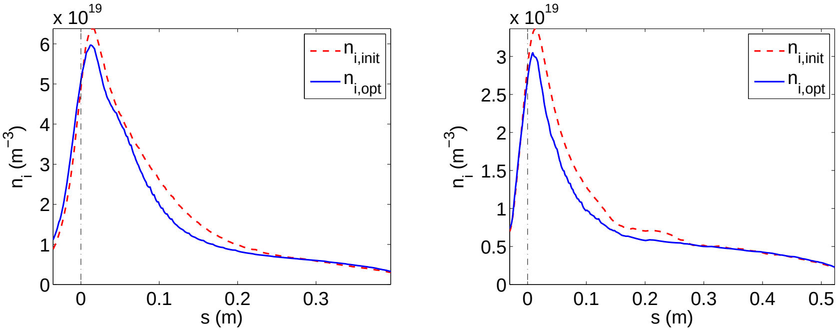


(c) Energy loss ionization (MW m^{-3}) (d) Energy loss ionization (MW m^{-3})

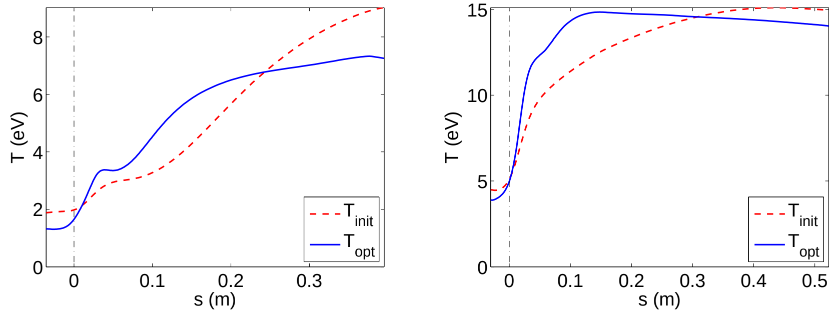


(e) Impurity radiation (MW m^{-3}) (f) Impurity radiation (MW m^{-3})

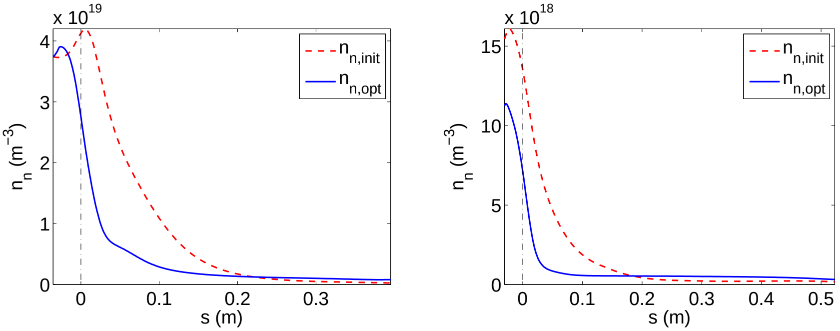
Figure 5.13: Comparison of the energy balances of initial (figures on the left) and optimized (figures on the right) target designs.



(a) Ion density, inner target (b) Ion density, outer target

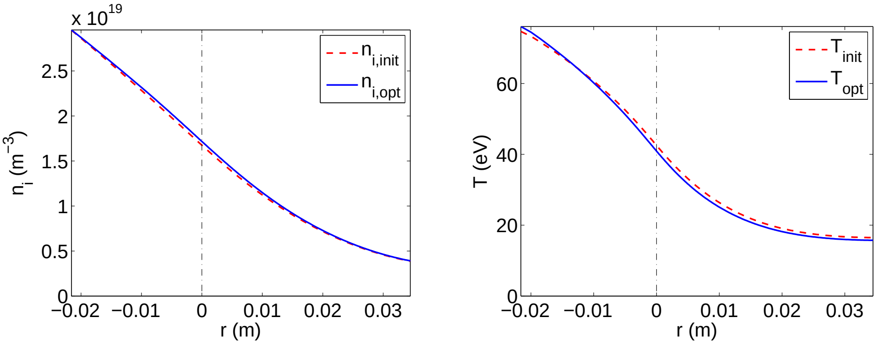


(c) Temperature, inner target (d) Temperature, outer target

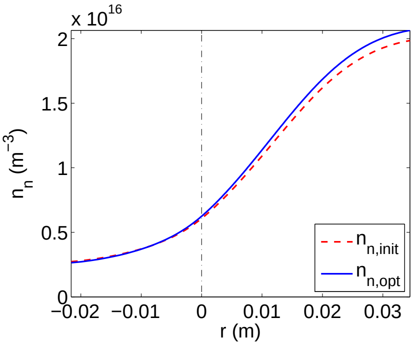


(e) Neutral density, inner target (f) Neutral density, outer target

Figure 5.14: Target profiles in initial and optimized configurations.



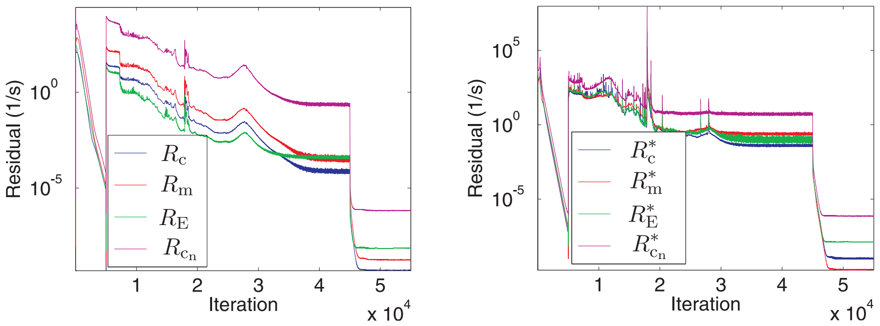
(a) Ion density (b) Temperature



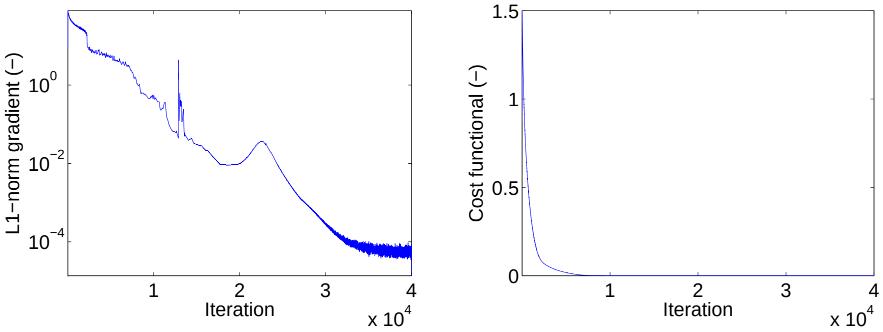
(c) Neutral density

Figure 5.15: Outer midplane profiles in initial and optimized configurations.

The residuals of state and adjoint equations, as well as the L1-norm of the gradient and the value of the cost functional are shown in Fig. 5.16. In the first 5.000 iterations, the initial state and adjoint equations were simulated. In order to save some computational time, the initial state was not converged to machine precision. Starting from the initial state, 40.000 one-shot iterations were performed. State and adjoint equations jointly converge, but the residuals hang at a level still above machine precision. The reason is in some small, local oscillatory behavior of a few design variables. However, the cost functional and shape sensitivity are significantly reduced. At the very end, an additional 2.500 iterations of the state and adjoint equations are sufficient to fully converge the equations. Taking into account that the stationary behavior of the residuals was already obtained after 32.500 one-shot iterations, the total cost of optimization was approximately 10 times the cost of a single simulation.



(a) State equations (b) Adjoint equations

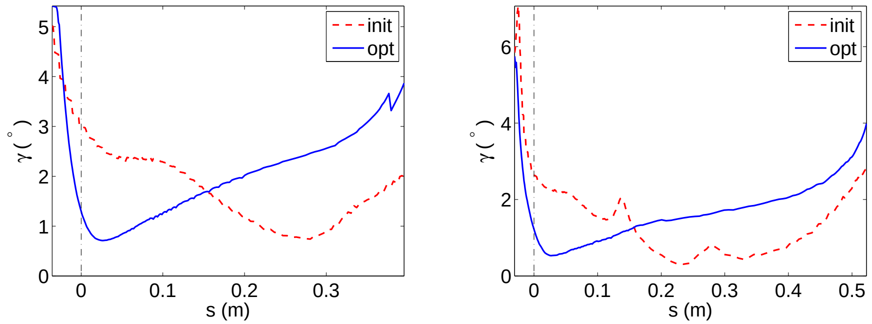


(c) L1-norm shape sensitivity (d) Cost functional

Figure 5.16: Evolution of the residuals during optimization.

Robustness and off-design performance

The targets of the optimized divertor presented in this section are more inclined with respect to the magnetic field than the initial targets. This increased tilting helps to spread out the energy over a larger area, thereby reducing the peak loads. In principle, the targets can be made almost tangent to the magnetic field. However, practical considerations limit the angle that can be safely used. Indeed, it has to be ensured that the heat load can still be handled in case of misalignment of a heat tile. If the angle is too shallow, misalignment may lead to the exposure of the edge of the tile, with failure of the tile as a likely result. Due to the magnetic field pitch, the angle of incidence γ between the target and the magnetic field is not the same as the one seen in the poloidal plane.



(a) Inner target (b) Outer target

Figure 5.17: Comparison of angle between target and magnetic field for initial and optimized cases.

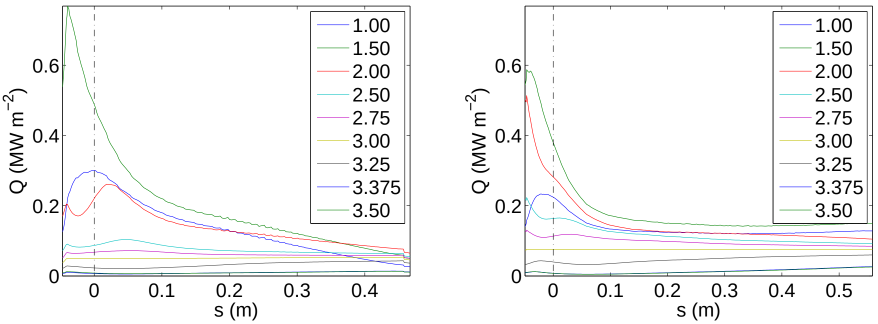
Some trigonometry quickly gives

$$\gamma = \sin^{-1}\left(b_\theta \cos\alpha\right), \tag{5.4}$$

where α is the angle between the surface normal and the poloidal projection of the magnetic field. Figure 5.17 compares the inclinations in initial and final configurations. The angle is indeed reduced near the strike points, but not to unacceptably low levels. If needed, a minimum incidence angle could also be taken as a design constraint in the optimization algorithm. The addition of such a nonlinear constraint would require careful treatment.

Other issues to be considered are the flexibility with respect to the magnetic equilibrium. Since the optimized divertor legs are much narrower compared to the initial configuration, the location of the separatrix strike points is more restricted. This may require more advanced control of the equilibrium. Furthermore, the closed geometry limits the freedom in magnetic equilibria.

The optimized uniform energy profiles at the targets have been obtained for a specific value of core power and density. It is now investigated how well the optimized design performs in off design points by repeating the density scan performed on the initial configuration. Figure 5.18 shows that the total energy fluxes to the targets stay relatively uniform for core densities in the range of $2.5 \cdot 10^{19}$ m^{-3} to $3.25 \cdot 10^{19}$ m^{-3} (and even higher core densities, but then the fluxes are very low due to the strong core radiation and detachment). Only at low densities the profiles are peaked again. Still, Figure 5.19 shows that over the entire density range the magnitude of the peak is reduced considerably compared to the initial configuration. This also seems to suggest that robust



(a) Inner target (b) Outer target

Figure 5.18: Total target energy fluxes as a function of core density n_c for optimized divertor configuration. The legend gives the value of n_c in 10^{19} m^{-3}.

design can be done quite effectively by considering the lowest density (and highest power) in the desired parameter range.

In Figure 5.20, the integrated ion and energy balances are compared for the initial and optimized divertors. The change in divertor design has lead to a reduction in all particle fluxes over the entire density range. Since core power is fixed, the energy fluxes have not changed much in this case. Qualitatively, the behavior of the initial and optimized reactors is the same. Roll-over still happens at approximately the same density. The radiating mantle, which is mainly determined by density and temperature in the closed field line region, is also largely unaffected.

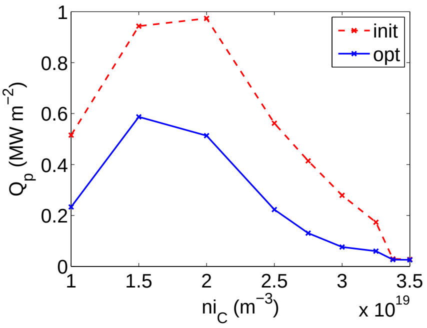
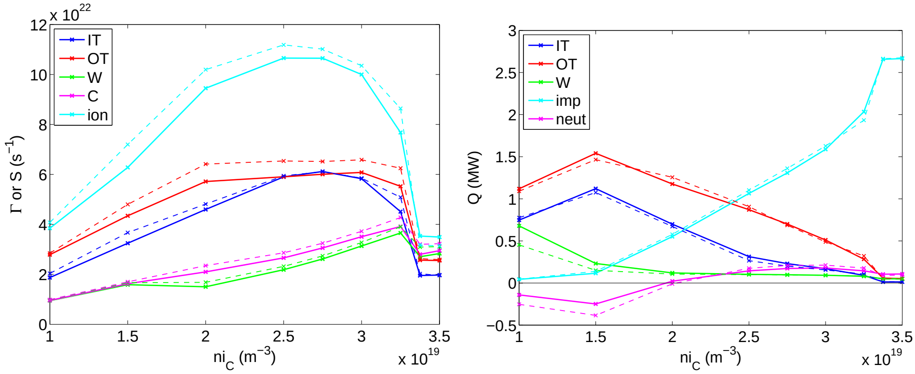


Figure 5.19: Peak target loads as a function of core density.



(a) Ion fluxes

(b) Energy fluxes

Figure 5.20: Integrated fluxes as a function of core density. Solid lines are for the optimized divertor, dashed lines for the initial configuration (cf. Fig. 5.7). IT: flux to inner target, OT: flux to outer target, W: flux to wall, C: flux from core, ion: total (net) volumetric ion source, imp: total energy sink due to impurity radiation, neut: total energy source due to neutral interactions. Fluxes to the private flux boundary are relatively small, and not shown in the figures.

5.4 Conclusion

In this chapter, optimal shape design is applied to realistic divertor configurations. Building on the state and adjoint solvers developed in this thesis, the main additional challenge is the handling of the strongly deformed geometries which may arise in reactor-scale devices. An efficient and reliable grid generator is essential in the framework of one-shot optimization. Since fully remeshing at every iteration would be too expensive, the grids are stretched along the field lines to match the deforming target surface. In this way, the cost of grid generation is negligible compared to the cost of an iteration of state or adjoint equations.

It is shown that accurate shape sensitivities can be computed with the velocity method. However, it is essential that the grid is sufficiently fine. Using the one-shot method, an optimized divertor design is obtained at a cost of approximately 10 state simulations.

The power of the algorithm is illustrated for a model problem which represents a scaled version of the ITER F57 equilibrium. In the optimized configuration, the integrated energy flux to the target is higher than in the initial configuration, while the peak load is removed. This is achieved by narrower V-shaped targets.

A density scan shows that the resulting divertor has a uniform energy load over a rather large density range. Furthermore, upstream conditions are not significantly altered.

Chapter 6

Optimal Shape Design for Coupled Fluid-Kinetic Systems

The design optimization problems studied in the previous chapters focus on fluid models for the edge plasma. In these models, the plasma and neutral behavior is described by 2D or 3D partial differential equations, which can be solved by classical finite volume discretization techniques. However, in practice these fluid models are often not sufficiently accurate. Especially for the neutral component, a kinetic description is to be preferred. Another important kinetic process in the edge plasma is radiation transport. Due to the high dimensionality, the kinetic equations are usually solved with a Monte Carlo procedure, resulting in coupled fluid-kinetic codes such as B2-EIRENE [96], EDGE2D-NIMBUS [113], EDGE2D-EIRENE [128], and UEDGE-DEGAS [101, 117]. While these models improve the accuracy of the simulations, the introduction of statistical noise due to the Monte Carlo codes also poses new challenges on for example the definition of convergence.

In this chapter, the aim is to investigate which additional difficulties are encountered for design optimization by including a kinetic neutral equation in the edge plasma model. First, the main elements of a kinetic neutral model are reviewed in Section 6.1. In order to avoid too much complexity in a first exploration step, this chapter is built around a strongly reduced 1D model for the edge plasma presented in Section 6.2 with a representative shape optimization problem discussed in Section 6.3. The adjoint kinetic equations are derived and analyzed in view of their solution with Monte Carlo methods in Section 6.4.

While the present chapter focuses on the theoretical derivation and analysis of the adjoint equations for a 1D model problem, Chapter 7 will study more in detail the implementation aspects, including the influence of statistical noise on the convergence of the coupled fluid-kinetic systems and its effect on (shape) sensitivity analysis and optimization. Furthermore, the methodology developed in this chapters can be extended to 2D kinetic neutral modeling and radiative transfer problems. The latter application will be treated in Chapter 8.

6.1 Kinetic Neutral Gas Modeling

6.1.1 Coupled Fluid-Kinetic Edge Plasma Model

Rather than using a simplified pressure diffusion relation, the aim is now to consider edge modeling problems in which the neutrals are described by a full kinetic equation. In practice, there may be several different types of neutrals in the edge plasma (atoms, molecules), chemical interactions between different species, inter-particle collisions, etc. Therefore, transport codes such as EIRENE solve a set of generalized nonlinear Boltzmann equations [95]. The starting point in this thesis is a reduced form of the Boltzmann equation encountered in linear transport theory, taking into account a single neutral atom, and assuming that neutrals only interact with the host medium, i.e. the plasma. In general, this is a 7 dimensional (linear) equation for the neutral distribution function $f_{\mathrm{n}}(\mathbf{r},\mathbf{v},t)$:

$$\begin{aligned}&\frac{\partial}{\partial t}f_{\mathrm{n}}(\mathbf{r},\mathbf{v},t)+\mathbf{v}\cdot\nabla_{\mathbf{r}}f_{\mathrm{n}}(\mathbf{r},\mathbf{v},t)+\Sigma_{\mathrm{t}}(\mathbf{r},\mathbf{v})\left|\mathbf{v}\right|f_{\mathrm{n}}(\mathbf{r},\mathbf{v},t)\\ &\qquad = S(\mathbf{r},\mathbf{v},t)+\int\Sigma_{\mathrm{t}}(\mathbf{r},\mathbf{v}')\left|\mathbf{v}'\right|f_{\mathrm{n}}(\mathbf{r},\mathbf{v}',t)C(\mathbf{r};\mathbf{v}'\to\mathbf{v})\,\mathrm{d}\mathbf{v}'. \end{aligned} \tag{6.1}$$

This equation can be interpreted as a continuity equation for particles in the space and velocity intervals $[\mathbf{r},\mathbf{r}+\mathrm{d}\mathbf{r}]$ and $[\mathbf{v},\mathbf{v}+\mathrm{d}\mathbf{v}]$. $S(\mathbf{r},\mathbf{v},t)$ is the source of particles at position $\mathbf{r}$ with velocity $\mathbf{v}$. $\Sigma_{\mathrm{t}}(\mathbf{r},\mathbf{v})$ is the total macroscopic cross-section (the inverse mean free path). The third term on the left hand side of the equation represents the total loss of particles from the phase space interval due to collisions of any kind. The collision integral, determined by the collision kernel $C(\mathbf{r};\mathbf{v}'\to\mathbf{v})$ describes the probability of a particle entering the phase space segment at $(\mathbf{r},\mathbf{v})$ due to a collision at $(\mathbf{r},\mathbf{v}')$. When different types of collisions are possible, each with macroscopic cross-section Σ_k, the collision kernel may be decomposed as

$$C(\mathbf{r};\mathbf{v}'\to\mathbf{v})=\sum_k p_k C_k(\mathbf{r};\mathbf{v}'\to\mathbf{v}), \tag{6.2}$$

where, by definition, $\Sigma_\mathrm{t}=\sum_k\Sigma_k$, and $p_k=\Sigma_k/\Sigma_\mathrm{t}$ is the probability of undergoing interaction k. c_k, defined as

$$c_k=\int C_k(\mathbf{r};\mathbf{v}'\to\mathbf{v})\,\mathrm{d}\mathbf{v} \tag{6.3}$$

is the number of secondary particles due to a collision at $(\mathbf{r},\mathbf{v}')$, and C_k/c_k is the conditional probability density of the post collision velocity $\mathbf{v}$ [115, 95]. If the number of particles is conserved in a collision, as is the case in for example charge exchange, $c_k=1$. In Eq. (6.1), external forces have been neglected. Therefore, particles move in straight lines between collisions.

Since the focus is on steady-state problems, the time dependence in Eq. (6.1) will be left out. Often, it is convenient to write the equation above in terms of the transport flux $\Phi_\mathrm{n}(\mathbf{r},\mathbf{v})\equiv|\mathbf{v}|\,f_\mathrm{n}(\mathbf{r},\mathbf{v})$, giving

$$\begin{aligned}\mathbf{s}\cdot\nabla_\mathbf{r}\Phi_\mathrm{n}(\mathbf{r},\mathbf{v})+\Sigma_\mathrm{t}(\mathbf{r},\mathbf{v})\Phi_\mathrm{n}(\mathbf{r},\mathbf{v})&\\ =\;\; S(\mathbf{r},\mathbf{v})+\int\Sigma_\mathrm{t}(\mathbf{r},\mathbf{v}')\Phi_\mathrm{n}(\mathbf{r},\mathbf{v}')C(\mathbf{r};\mathbf{v}'\to\mathbf{v})\,\mathrm{d}\mathbf{v}',&\end{aligned} \tag{6.4}$$

with $\mathbf{s}=\mathbf{v}/|\mathbf{v}|$ the unit vector in the direction of particle motion.

This kinetic equation for the neutrals now replaces the pressure diffusion model of Eq. (3.15), and is solved in a coupled way with the plasma fluid equations (3.5), (3.7), (3.9), and (3.21) discussed in Chapter 3. The latter equations (in steady-state form) are repeated here for convenience:

$$\nabla\cdot(n\mathbf{V}_\mathrm{i}) = S_{n_\mathrm{i}}, \tag{6.5}$$

$$\nabla\cdot\left(mnu_{||}\mathbf{V}_\mathrm{i}-\eta^\mathrm{i}\nabla u_{||}\right) = S_{mu_{||}}-\nabla_{||}p, \tag{6.6}$$

$$nu_r = -D^\mathrm{i}\nabla_r n, \tag{6.7}$$

$$\nabla\cdot\left(\frac{5}{2}(1+Z_\mathrm{i})\mathbf{\Gamma}^\mathrm{i}T-\kappa\nabla T\right) = S_E+\nabla u_{||}\cdot\eta^\mathrm{i}\cdot\nabla u_{||}+u_{||}\nabla_{||}p. \tag{6.8}$$

The ion continuity and momentum equations are the same as before. The energy equation requires some further clarification. Since the kinetic equation (6.4) describes the whole neutral distribution function, it encompasses neutral continuity, momentum and energy equations. Therefore, neutral energy transport should no longer be included in the energy equation (6.8), which is now reduced to a plasma (ion and electron) energy equation for the ion and electron temperature $T = T_\mathrm{i} = T_\mathrm{e}$. The neutral temperature can be deduced from f_n, and is not necessarily equal to T.

In accordance to the model presented in Chapter 3, the ion-neutral interactions include (electron impact) ionization, charge exchange, and (radiative) recombination. Whereas the rate coefficients for these interactions (specified in Section 3.1.6) determined the pressure diffusion coefficient and weighted neutral density before, see Eqs. (3.13) and (3.14), they now appear in the macroscopic cross-sections (and thus also the collision kernel),

$$\Sigma_\mathrm{i} = \frac{n_\mathrm{e} K_\mathrm{i}}{|\mathbf{v}|}, \qquad \Sigma_\mathrm{cx} = \frac{n_\mathrm{i} K_\mathrm{cx}}{|\mathbf{v}|}. \tag{6.9}$$

The total cross-section is $\Sigma_\mathrm{t} = \Sigma_\mathrm{i} + \Sigma_\mathrm{cx}$. Only the charge-exchange cross-section will contribute to the collision kernel, because the neutral particle is lost at ionization, so $C_\mathrm{i}(\mathbf{r}; \mathbf{v}' \to \mathbf{v}) = 0$. Therefore,

$$C(\mathbf{r}; \mathbf{v}' \to \mathbf{v}) = \frac{\Sigma_\mathrm{cx}}{\Sigma_\mathrm{t}} C_\mathrm{cx}(\mathbf{r}; \mathbf{v}' \to \mathbf{v}).$$

Since volume recombination does not involve neutral particles as collision partners, there is no corresponding macroscopic cross-section. This process appears as part of the neutral source S:

$$\int S(\mathbf{r}, \mathbf{v}) \, \mathrm{d}\mathbf{v} = K_\mathrm{r} n_\mathrm{i} n_\mathrm{e}.$$

Source terms in the plasma equations involve integrals across the neutral distribution function,

$$S_{n_\mathrm{i}} = \int \Phi_\mathrm{n} \Sigma_\mathrm{i} \, \mathrm{d}\mathbf{v} - K_\mathrm{r} n_\mathrm{i} n_\mathrm{e}, \tag{6.10}$$

$$S_{mu_{||}} = \int \Phi_\mathrm{n} (m(\mathbf{v} \cdot \mathbf{e}_{||}) \Sigma_\mathrm{t} - m u_{||} \Sigma_\mathrm{cx}) \, \mathrm{d}\mathbf{v} - m K_\mathrm{r} n_\mathrm{i} n_\mathrm{e} u_{||}, \tag{6.11}$$

$$S_E = S_{E_\mathrm{t}} + \frac{m u_{||}^2}{2} S_{n_\mathrm{i}} - u_{||} S_{mu_{||}}, \tag{6.12}$$

$$\begin{aligned} S_{E_\mathrm{t}} = {} & \int \Phi_\mathrm{n} \Sigma_\mathrm{i} \left(\frac{m\mathbf{v}^2}{2} - E_\mathrm{i} \right) \mathrm{d}\mathbf{v} - K_\mathrm{r} n_\mathrm{i} n_\mathrm{e} \left(\frac{m u_{||}^2}{2} + 3T \right) \\ & + \int \Phi_\mathrm{n} \Sigma_\mathrm{cx} \left(\frac{m\mathbf{v}^2}{2} - \frac{m u_{||}^2}{2} - \frac{3}{2} T \right) \mathrm{d}\mathbf{v} - c_z n_\mathrm{i} n_\mathrm{e} L_z. \end{aligned} \tag{6.13}$$

As before, E_i is the energy provided by the plasma at ionization. Since the radial ion velocity is approximated with an anomalous diffusive model and the diamagnetic ion velocity is neglected, only $u_{||}^2$ contributes to the internal energy source. The first two integrals reduce to the corresponding expressions given in Chapter 3 by using

$$n_\mathrm{n} \equiv \int f_\mathrm{n} \, \mathrm{d}\mathbf{v}, \qquad n_\mathrm{n} \mathbf{V}_\mathrm{n} \equiv \int f_\mathrm{n} \mathbf{v} \, \mathrm{d}\mathbf{v}, \tag{6.14}$$

and assuming that the rate coefficients K_i and K_cx are independent of $\mathbf{v}$. The two last terms in Eq. (6.12) arise due to the conversion of the total energy source S_{E_t} to the internal energy source S_E, see Eq. (3.20) and Ref. [25]. This internal energy source is not identical to the one from the fluid model, because the neutrals are now treated separately. Elaboration of this source using

$$\frac{3}{2} n_\mathrm{n} T_\mathrm{n} \equiv \int f_\mathrm{n} \frac{m\mathbf{w}^2}{2} \, \mathrm{d}\mathbf{v}, \tag{6.15}$$

where $\mathbf{w} \equiv \mathbf{v} - \mathbf{V}_\mathrm{n}$ is the neutral thermal velocity, and again assuming rate coefficients are independent of $\mathbf{v}$ leads to

$$\begin{aligned} S_E = {} & \frac{m}{2} \left(\left(u_{||} - u_{\mathrm{n}||} \right)^2 + u_{\mathrm{n}r}^2 + u_{\mathrm{n}\perp}^2 \right) \left(K_\mathrm{i} n_\mathrm{e} n_\mathrm{n} + K_\mathrm{cx} n_\mathrm{i} n_\mathrm{n} \right) \\ & + \frac{3}{2} K_\mathrm{i} n_\mathrm{e} n_\mathrm{n} T_\mathrm{n} + \frac{3}{2} K_\mathrm{cx} n_\mathrm{i} n_\mathrm{n} \left(T_\mathrm{n} - T \right) \\ & - 3 K_\mathrm{r} n_\mathrm{e} n_\mathrm{i} T - E_\mathrm{i} n_\mathrm{e} n_\mathrm{n} K_\mathrm{i} - c_z n_\mathrm{i} n_\mathrm{e} L_z. \end{aligned}$$

Note that by definition, $\int f_\mathrm{n}\mathbf{w}\,\mathrm{d}\mathbf{v} = 0$. Furthermore, $u_{\mathrm{n}||} = \mathbf{V}_\mathrm{n}\cdot\mathbf{e}_{||}$, $u_{\mathrm{n}r} = \mathbf{V}_\mathrm{n}\cdot\mathbf{e}_r$ and $u_{\mathrm{n}\perp} = \mathbf{V}_\mathrm{n}\cdot\mathbf{e}_\perp$.

Before moving on to the implementation of the kinetic neutral equation, consistency of the kinetic equation and the neutral continuity equation (3.10) is illustrated by integrating equation (6.4) over all velocities (i.e. by taking the zeroth order velocity moment of the equation). Integrating the first term in (6.4) gives

$$\int \mathbf{s}\cdot\nabla_\mathbf{r}\Phi_\mathrm{n}(\mathbf{r},\mathbf{v})\,\mathrm{d}\mathbf{v} = \nabla\cdot\left(n_\mathrm{n}\mathbf{V}_\mathrm{n}\right).$$

The subscript $\mathbf{r}$ of the divergence operator on the right hand side of the equation is dropped, because this term does not depend on $\mathbf{v}$ anymore. Combined, the other three terms in the kinetic equation lead to the total neutral source $S_{n_\mathrm{n}} = S_{n_\mathrm{n},1} + S_{n_\mathrm{n},2} + S_{n_\mathrm{n},3}$:

$$\begin{aligned}
S_{n_\mathrm{n},1} &= -\int \Sigma_\mathrm{t}(\mathbf{r},\mathbf{v})\Phi_\mathrm{n}(\mathbf{r},\mathbf{v})\,\mathrm{d}\mathbf{v},\\
S_{n_\mathrm{n},2} &= \int S(\mathbf{r},\mathbf{v})\,\mathrm{d}\mathbf{v} \;=\; K_\mathrm{r}n_\mathrm{i}n_\mathrm{e},\\
S_{n_\mathrm{n},3} &= \int\!\!\int \Sigma_\mathrm{t}(\mathbf{r},\mathbf{v}')\Phi_\mathrm{n}(\mathbf{r},\mathbf{v}')C(\mathbf{r};\mathbf{v}'\to\mathbf{v})\,\mathrm{d}\mathbf{v}'\,\mathrm{d}\mathbf{v}\\
&= \int \Sigma_\mathrm{cx}(\mathbf{r},\mathbf{v})\Phi_\mathrm{n}(\mathbf{r},\mathbf{v})\,\mathrm{d}\mathbf{v}.
\end{aligned}$$

Thus, the neutral continuity equation is recovered,

$$\nabla\cdot\left(n_\mathrm{n}\mathbf{V}_\mathrm{n}\right) = S_{n_\mathrm{n}}. \tag{6.16}$$

Of course, total conservation of mass of the coupled system is still satisfied: $S_{n_\mathrm{n}} = -S_{n_\mathrm{i}}$. The typical closure problem arises. In order to solve equation (6.16) for the neutral density n_n, the neutral velocity $\mathbf{V}_\mathrm{n}$ must be known, which results from the first order moment of the kinetic equation (the momentum equation). The momentum equation in turn will depend on second order moments and so on. This procedure has to be cut off at some point, typically after the energy equation. A closure scheme is introduced to model the remaining unknown higher order moments and retain a solvable set of PDEs. In the closure scheme, some approximations and assumptions have to be made, and these determine the validity range of the resulting PDE model. However, in the current application, the neutral kinetic equation is solved as part of the coupled edge plasma model, so these higher order moments can be *computed.* This observation will prove very useful in accelerating convergence of the coupled fluid-kinetic system by introducing *short-cycle iterations* based on consistent reduced models. This will be illustrated for a 1D model in Chapter 7, Section 7.2.4.

It is remarked that while the behavior of the neutrals is described by a complex 6D (7D) equation (6.1), its complete solution f_{n} or Φ_{n} contains too much detail and is often not required in practical problems. For example, for the solution of the plasma field equations (6.5)–(6.8), only the source terms (6.10)–(6.12) are needed. These source terms are averages over the neutral distribution function and therefore only 3D quantities (or 2D in case of toroidal symmetry). Other quantities of interest or *responses*, such as neutral density, velocity and temperature (Eqs. (6.14)–(6.15)) and neutral fluxes to plasma-facing components are again velocity space averages. In general, these responses can be written as integrals over phase space in terms of *detector functions* $g(\mathbf{r}, \mathbf{v})$,

$$\langle \Phi_{\mathrm{n}} \, , \, g \rangle \equiv \int\!\!\int \Phi_{\mathrm{n}}(\mathbf{r}, \mathbf{v}) g(\mathbf{r}, \mathbf{v}) \, \mathrm{d}\mathbf{v} \, \mathrm{d}\mathbf{r}.$$

The expression above defines the angle brackets as an inner product in phase space. If point wise information at $\mathbf{r} = \mathbf{r}_0$ is needed, $g(\mathbf{r}, \mathbf{v})$ can include a Dirac-delta $\delta(\mathbf{r} - \mathbf{r}_0)$. In combination with a (finite volume) fluid code, cell averaged sources are needed. For example, in order to compute the ionization source in a particular cell of the fluid code, the detector function is

$$g(\mathbf{r}, \mathbf{v}) = \chi(\mathbf{r}) \Sigma_{\mathrm{i}}(\mathbf{r}, \mathbf{v}),$$

where the characteristic function $\chi(\mathbf{r})$ is 1 inside the cell and 0 outside the cell. In practice, there are often several thousands of detector functions, e.g. particle, momentum and energy sources in every cell, the neutral density in every cell, etc. This specific problem structure is very well suited for solution with a Monte Carlo method, since such a code can accumulate contributions to all the different responses during the simulation of individual particles, while the complete neutral distribution doesn't have to be known or stored.

6.1.2 Solution of the Kinetic Equation with a Monte Carlo Procedure

The solution of the kinetic equation (6.4) with a Monte Carlo procedure means a direct simulation of the underlying kinetic processes by following the trajectory of a large number of individual neutral particles n_{t}. The source in the kinetic equation determines where the particles are launched and what their initial distributions is. Then, the particles fly a certain distance, collide with the background plasma, change flight direction and so on until they are either ionized or absorbed by a boundary. The contribution of these individual

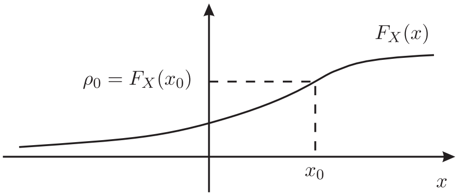


Figure 6.1: Sampling from a general cumulative distribution function. Figure based on Ref. [115].

particles to the responses $\langle \Phi_\mathrm{n}\,,\, g\rangle$ is accumulated during their lifetime. Here, only a brief description of the main elements of a Monte Carlo neutral code is given. Details on the methods used can be found in Refs. [115, 95].

At the basis of the Monte Carlo code is a (pseudo-) random number generator[1]. This generator can draw independent random numbers $0 \le \rho \le 1$ from a uniform probability density function $h(x)$ on the interval $[0, 1]$ [115]:

$$h(x) = \begin{cases} 1 & 0 \le x \le 1, \\ 0 & \text{otherwise.} \end{cases}$$

At every event in the life of an individual particle, such a random number will determine what happens to the particle. Therefore, a random variable X is associated with each event, and the random number ρ has to be related to the (cumulative) distribution function $F_X(x)$ of this random variable. $F_X(x) = P(X \le x) = \int_{-\infty}^{x} f(t)\mathrm{d}t$ is the probability that X takes a value less than or equal to x, and $f(x)$ is the probability density function. It can be shown that the distribution function is sampled correctly by associating with the random number ρ_0 the value x_0 so that

$$\rho_0 = F_X(x_0), \tag{6.17}$$

see Fig. 6.1. This is a very powerful result, since it allows to sample any process correctly if the distribution of the process is known.

In the kinetic neutral model described above, distributions are required for the flight distance to the next event (F_d), and for the type of collision the particle will experience (F_c). The distribution of the flight distance is derived from the definition of the total cross-section Σ_t. Indeed, along its flight direction $\mathbf{s}$ a neutral beam is attenuated as

$$\frac{\mathrm{d}}{\mathrm{d}l}\Phi_\mathrm{n}(\mathbf{r} + l\mathbf{s}, \mathbf{v}) = -\Sigma_\mathrm{t}(\mathbf{r} + l\mathbf{s}, \mathbf{v})\Phi_\mathrm{n}(\mathbf{r} + l\mathbf{s}, \mathbf{v}), \tag{6.18}$$

---

[1] In a computer, there is always some deterministic algorithm at the basis of a random number generator. Therefore, the numbers are not truly random, and are called *pseudorandom*. For ease, they will just be called random numbers in what follows.

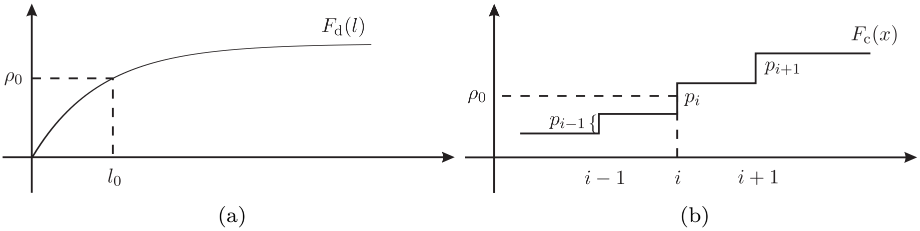


Figure 6.2: Sampling from (a) cumulative flight distance distribution and (b) collision distribution. Figure based on Ref. [115].

with l the flight distance. Therefore, $\Sigma_\mathrm{t}(\mathbf{r}+l\mathbf{s},\mathbf{v})\mathrm{d}l$ is the probability of having any type of interaction in the interval $\mathrm{d}l$ around l. By integrating Eq. (6.18) along the flight direction, the probability that a particle travels a distance l without having any type of collision is found as $\exp\left(-\int_0^l \Sigma_\mathrm{t}(\mathbf{r}+t\mathbf{s})\mathrm{d}t\right)$. Combining results, the probability $P_\mathrm{d}(l)$ of having the next interaction in the interval $[l, l+\mathrm{d}l]$ is

$$f_\mathrm{d}(l) = \Sigma_\mathrm{t}(\mathbf{r}+l\mathbf{s})\exp\left(-\int_0^l \Sigma_\mathrm{t}(\mathbf{r}+t\mathbf{s})\mathrm{d}t\right)\mathrm{d}l,$$

giving the cumulative distribution for the flight distance

$$F_\mathrm{d}(l) = \int_0^l \Sigma_\mathrm{t}(\mathbf{r}+u\mathbf{s})\exp\left(-\int_0^u \Sigma_\mathrm{t}(\mathbf{r}+t\mathbf{s})\mathrm{d}t\right)\mathrm{d}u. \tag{6.19}$$

For the simple case where Σ_t is constant, this reduces to $F_\mathrm{d}(l) = 1-\exp(-\Sigma_\mathrm{t} l)$, Fig. 6.2 (a).

In contrast to the flight distance distribution (6.19), the distribution for the type of collision is a discrete one. The possible collisions may be numbered according to some index k, with $1 \le k \le K$ and K the number of different collisions considered. As discussed earlier, the probability of a collision of type k occurring is $p_k = \Sigma_k/\Sigma_\mathrm{t}$. Since $\Sigma_\mathrm{t} = \sum_k \Sigma_k$, it also holds that $\sum_k p_k = 1$. Therefore, the cumulative distribution can be written as [115]

$$F_\mathrm{c}(x) = \sum_{k\le x} p_k, \tag{6.20}$$

see Fig. 6.2 (b). After drawing a random number ρ, collision k will occur if

$$\sum_{i=0}^{k-1} p_i < \rho \le \sum_{i=0}^{k} p_i,$$

with $p_0 = 0$.

Computing a response of the form $\langle \Phi_\mathrm{n} \, , \, g \rangle$ with Monte Carlo is based on generating a number of *trajectories*, *histories* or *chains* $\omega^n = (x_0, x_1, x_2, ..., x_n)$ of the random walk process. These trajectories are a sequence of states $x_i = (\mathbf{r}_i, \mathbf{v}_i)$ in velocity space visited by the particle during its life. The length of the chain itself is also a random variable. For each response, a random variable or *estimator* $\xi(\omega)$ is needed with expected value $\mu_\xi = \langle \Phi_\mathrm{n} \, , \, g \rangle$. It can be shown that the so-called *track-length estimator*

$$\xi^\mathrm{t}(\omega) = \sum_{i=0}^{n-1} \left(\int_0^{\mathbf{s}_i \cdot (\mathbf{r}_{i+1} - \mathbf{r}_i)} g(\mathbf{r}_i + l\mathbf{s}_i, \mathbf{v}_i) \mathrm{d}l \right), \tag{6.21}$$

with $\mathbf{s}_i = \mathbf{v}_i / |\mathbf{v}_i|$, is such an unbiased estimator for $\langle \Phi_\mathrm{n} \, , \, g \rangle$ [115]. This means that the track-length estimator has the same expected value as the response itself, and thus its average over many trajectories can be seen as an approximation to the response,

$$\langle \Phi_\mathrm{n} \, , \, g \rangle \approx \overline{\xi^\mathrm{t}}_{n_\mathrm{t}} = \frac{1}{n_\mathrm{t}} \sum_{i=1}^{n_\mathrm{t}} \xi^\mathrm{t}(\omega_i). \tag{6.22}$$

Furthermore, since the trajectories are independent and equally distributed, the Central Limit Theorem applies. This theorem states that if ξ_1, ξ_2,... are independent and identically distributed random variables with mean μ and standard deviation σ, then the average over N samples

$$\bar{\xi}_N = \frac{1}{N} \sum_{i=1}^{N} \xi_i$$

is asymptotically normal with mean μ and standard deviation $\sigma/\sqrt{N}$, or

$$\lim_{N \to \infty} P\left(\frac{\bar{\xi}_N - \mu}{\sigma/\sqrt{N}} \leq x \right) = \phi(x), \tag{6.23}$$

with $\phi(x)$ the standard normal distribution [115]. Therefore, the standard deviation or statistical uncertainty on the response $\langle \Phi_\mathrm{n} \, , \, g \rangle$ will decrease as $n_\mathrm{t}^{-1/2}$. From this result, the trade-off between computational time and accuracy of Monte Carlo solvers becomes clear. Indeed, the computational time scales linearly with the number of simulated trajectories n_t, while the accuracy improves only with a square root dependency. On the other hand, a (linear) Monte Carlo code can be parallelized quite easily, since all trajectories are independent.

6.2 A 1D Kinetic Neutral Model

In order to assess the behavior of fluid-kinetic code coupling and to analyze in detail the adjoint kinetic equations, a simplified 1D model is constructed. In particular, the kinetic neutral equation will be significantly reduced. This will allow solving the kinetic equation also with finite volume techniques, providing a strong benchmark for the Monte Carlo codes and a thorough assessment of performance and accuracy.

A simple 1D domain in the poloidal plane along a ‘straightened out’ field line with constant pitch b_θ is considered. This reduced model problem serves to describe the divertor behavior from an upstream position ($\theta = 0$) to the divertor target ($\theta = L_\theta$). Only the ion continuity and momentum equations are included in the model, coupled to a kinetic equation for the neutrals:

$$\frac{\partial}{\partial\theta}\left(nu_\theta\right) = S_{n_\mathrm{i}} - C_n n, \tag{6.24}$$

$$\frac{\partial}{\partial\theta}\left(mnu_\theta u_{||} - \eta_\theta^\mathrm{i}\frac{\partial u_{||}}{\partial\theta}\right) = S_{mu_{||}} - b_\theta\frac{\partial p}{\partial\theta}, \tag{6.25}$$

$$\mathbf{s}\cdot\mathbf{e}_\theta\frac{\partial\Phi_\mathrm{n}}{\partial\theta} + \Sigma_\mathrm{t}\Phi_\mathrm{n} = S + \int\Sigma_\mathrm{t}\Phi_\mathrm{n}(\theta,\mathbf{v}')C(\mathbf{v}'\to\mathbf{v})\,\mathrm{d}\mathbf{v}'. \tag{6.26}$$

The ion and electron temperatures are constant along the field line. Furthermore, mono-energetic neutrals are assumed, which can only move with thermal speed $v_{n_\mathrm{n},\mathrm{th}}$ in the poloidal plane along the poloidal projection of the field line, i.e. only in the directions $\mathbf{e}_\theta$ and $-\mathbf{e}_\theta$. Their thermal speed is computed as $v_{n_\mathrm{n},\mathrm{th}} = (2T_\mathrm{n}/m)^{1/2}$. However, in order to ensure that neutrals stay mono-energetic the neutral temperature T_n is an input parameter in this model. It is not computed self-consistently from Eq. (6.15). The ion sink $-C_n n$ represents a loss term due to anomalous radial diffusion, $C_n \sim D^\mathrm{i}/\lambda_n^2$.

Volume recombination only becomes important at temperatures of the order of 1 eV, so it is neglected here. This means that the source S is zero, except at the target where the neutrals are born. Neutrals and ions interact through ionization and charge exchange only, with macroscopic cross-sections found by using $|\mathbf{v}| = v_{n_\mathrm{n},\mathrm{th}}$ in Eq. (6.9). The total macroscopic cross-section is $\Sigma_\mathrm{t} = \Sigma_\mathrm{i} + \Sigma_\mathrm{cx}$. It is observed that the cross-sections depend on local plasma parameters, in this case on the density n, and thus vary with position. However, they are independent of the velocity coordinate $\mathbf{v}$. For the collision kernel,

$$C(\theta;\mathbf{v}'\to\mathbf{v}) = \frac{\Sigma_\mathrm{cx}}{\Sigma_\mathrm{t}}\left(\frac{1}{2}\delta(\mathbf{s}-\mathbf{s}') + \frac{1}{2}\delta(\mathbf{s}+\mathbf{s}')\right) \tag{6.27}$$

is taken. This means that after a charge-exchange collision — which occurs with probability $p_{\mathrm{cx}} = \Sigma_{\mathrm{cx}}/\Sigma_{\mathrm{t}}$ — neutrals have equal probability of moving in the same or reverse direction as prior to the collision, still with speed $v_{n_{\mathrm{n}},\mathrm{th}}$. This kernel ensures that neutrals will keep moving in either the positive or negative θ-directions ($\mathbf{e}_\theta$ resp. $-\mathbf{e}_\theta$) after a collision.

To conclude this section, it is remarked that simplified spatially 1D kinetic neutral models have been used in the past to estimate neutral distributions at plasma-facing components, see for example Ref. [94]. Usually, a homogeneous plasma background was assumed in these models, allowing to solve the kinetic equation analytically. In this thesis, the plasma model is somewhat more elaborate, but the neutrals are oversimplified by taking them as mono-energetic. However, the motivation to use such model in this chapter is not in its physical relevance, but in its value as 'numerical play ground'. The model contains some important elements that make the simulation of the edge plasma complex (e.g. fluid-kinetic interactions, nonlinearities, competition between parallel and radial flow), but is simple enough to allow decoupling the model into equations that can be solved with a finite volume rather than with a Monte Carlo approach. This is demonstrated in the next paragraphs.

Two-equation kinetic formulation

The main motivation of using such restricted neutral model is that it allows to decompose the kinetic equation into two convective equations with source terms. Indeed, due to the assumption of mono-energetic neutrals with velocity restricted to lie along $\mathbf{e}_\theta$ or $-\mathbf{e}_\theta$, $f_{\mathrm{n}}(\theta, \mathbf{v})$ and $\Phi_{\mathrm{n}}(\theta, \mathbf{v})$ can also be written as

$$\begin{aligned} f_{\mathrm{n}}(\theta,\mathbf{v}) &= f_{\mathrm{n},+}(\theta)\delta(\mathbf{s}-\mathbf{e}_\theta) + f_{\mathrm{n},-}(\theta)\delta(\mathbf{s}+\mathbf{e}_\theta), \\ \Phi_{\mathrm{n}}(\theta,\mathbf{v}) &= \Phi_{\mathrm{n},+}(\theta)\delta(\mathbf{s}-\mathbf{e}_\theta) + \Phi_{\mathrm{n},-}(\theta)\delta(\mathbf{s}+\mathbf{e}_\theta), \end{aligned}$$

where the functions $f_{\mathrm{n},+}$, $f_{\mathrm{n},-}$, $\Phi_{\mathrm{n},+}$, and $\Phi_{\mathrm{n},-}$ only depend on the spatial coordinate. Plugging this into the kinetic equation (6.26), together with the kernel (6.27), and grouping terms in $\delta(\mathbf{s}-\mathbf{e}_\theta)$ and $\delta(\mathbf{s}+\mathbf{e}_\theta)$ leads to the two equations

$$\left\{ \begin{aligned} \frac{\partial \Phi_{\mathrm{n},+}}{\partial \theta} + \Sigma_{\mathrm{t}}\Phi_{\mathrm{n},+} &= S_+ + \frac{\Sigma_{\mathrm{cx}}}{2}(\Phi_{\mathrm{n},+} + \Phi_{\mathrm{n},-}), \\ -\frac{\partial \Phi_{\mathrm{n},-}}{\partial \theta} + \Sigma_{\mathrm{t}}\Phi_{\mathrm{n},-} &= S_- + \frac{\Sigma_{\mathrm{cx}}}{2}(\Phi_{\mathrm{n},+} + \Phi_{\mathrm{n},-}). \end{aligned} \right. \tag{6.28}$$

Also the volumetric source term has been split into contributions of neutrals moving in positive (S_+) and negative (S_-) directions. Both are zero in this case. These two convective equations can be implemented very easily with a finite volume approach, leading to interesting options for numerical validation of the Monte Carlo procedures.

Plasma source terms

In this simplified model, the following source terms appear in the ion continuity and momentum equations:

$$S_{n_\mathrm{i}} = \int \Phi_\mathrm{n} \Sigma_\mathrm{i} \, \mathrm{d}\mathbf{v}, \tag{6.29}$$

$$S_{mu_{||}} = \int \Phi_\mathrm{n} (m(\mathbf{v} \cdot \mathbf{e}_{||})\Sigma_\mathrm{t} - mu_{||}\Sigma_\mathrm{cx}) \, \mathrm{d}\mathbf{v}. \tag{6.30}$$

Since neutrals move only in the positive or negative θ directions, these integrals can be worked out, and yield the convenient form

$$\begin{aligned} S_{n_\mathrm{i}} &= \Sigma_\mathrm{i} \left(\Phi_{\mathrm{n},+} + \Phi_{\mathrm{n},-}\right), \\ S_{mu_{||}} &= mb_\theta v_{n_\mathrm{n},\mathrm{th}} \Sigma_\mathrm{t} \left(\Phi_{\mathrm{n},+} - \Phi_{\mathrm{n},-}\right) - mu_{||}\Sigma_\mathrm{cx} \left(\Phi_{\mathrm{n},+} + \Phi_{\mathrm{n},-}\right). \end{aligned}$$

Boundary conditions

Upstream, at $\theta = 0$, plasma density and particle flux (and thus parallel velocity) are fixed:

$$\left\{ \begin{aligned} n &= n_0, \\ nu_\theta &= \Gamma_{n,0}. \end{aligned} \right. \tag{6.31}$$

At the target, at $\theta = L_\theta$, sheath conditions are applied:

$$u_{||} = c_\mathrm{s}, \tag{6.32}$$

with $c_\mathrm{s} = \sqrt{\frac{T_\mathrm{i} + T_\mathrm{e}}{m}}$.

For the kinetic equation, boundary conditions are required for all *ingoing* directions $\mathbf{s}$. This is also nicely translated to the two-equation kinetic system (6.28), which requires boundary conditions for $\Phi_{\mathrm{n},+}$ at $\theta = 0$ and for $\Phi_{\mathrm{n},-}$ at $\theta = L_\theta$. The neutral source at the target is specified by a recycling coefficient $0 \leq R \leq 1$. This can either be interpreted as a localized source,

$$S(\theta, \mathbf{v}) = \delta(\theta - L_\theta)\delta(\mathbf{s} + \mathbf{e}_\theta) R b_\theta u_{||}(L_\theta) n(L_\theta),$$

or as part of the boundary conditions. At the core, neutrals are perfectly absorbed and no neutrals enter, while at the targets perfect reflection is assumed. Combined with the neutral source, this implies the following boundary conditions:

$$\left\{\begin{array}{lcll} \Phi_{\mathrm{n},+} & = & 0 & \text{at} \quad \theta = 0, \\ \Phi_{\mathrm{n},-} & = & R b_\theta u_{||} n + \Phi_{\mathrm{n},+} & \text{at} \quad \theta = L_\theta. \end{array}\right. \tag{6.33}$$

6.3 Optimization of Target Flux

The aim of this chapter is to set up a divertor shape optimization problem in the context of kinetic neutral modeling. To this end, a simple 1D analogue of the target energy load optimization problem is constructed. The goal is now to achieve a certain desired ion flux[2] Γ_d at the target by controlling only the length L_θ of the domain. An additional penalty term may be added to keep the domain length close to a desired length L_0. This objective is reflected in the following cost functional:

$$\begin{aligned} J(\Omega, \mathbf{q}) &= \frac{1}{2} \left(\Gamma - \Gamma_\mathrm{d}\right)^2 \Big|_{L_\theta} + \frac{\lambda}{2} \left(L_\theta - L_0\right)^2 \\ &= \frac{1}{2} \left(n u_\theta - \Gamma_\mathrm{d}\right)^2 \Big|_{L_\theta} + \frac{\lambda}{2} \left(L_\theta - L_0\right)^2 . \end{aligned} \tag{6.34}$$

This cost functional is to be regarded as a 'boundary integral' along the target. As before, the state equations and boundary conditions, Eqs. (6.24)–(6.26) and (6.31)–(6.33) form the state constraints in the optimization problem. The vector of state variables is $\mathbf{q} = (n, u_{||}, \Phi_\mathrm{n})^T$.

In order to formally eliminate the constraints, the state equations and boundary conditions are added to the cost functional by using Lagrange multipliers or adjoint state variables, $\mathbf{q}^* = (n^*, u_{||}^*, \Phi_\mathrm{n}^*)^T$ and $\mathbf{p}^* = (n_S^*, u_{||,S}^*, {\Phi_\mathrm{n}}_{,S}^*)^T$. Since the kinetic equation is defined in *phase space*, the corresponding adjoint variable will also be a phase space variable, and the equation has to be integrated over both spatial and velocity coordinates in the Lagrangian[3]:

$$L(\Omega, \mathbf{q}, \mathbf{q}^*) = J(\Omega, \mathbf{q}) + \left(\mathbf{q}^*, \boldsymbol{\mathcal{B}}(\Omega, \mathbf{q})\right), \tag{6.35}$$

---

[2] Since all temperatures are assumed to be constant along the flux tube, the energy flux is proportional to the ion particle flux.

[3] It is recalled that $\mathbf{q}^*$ is used to denote both the complete set of adjoint variables defined on domain and boundary, $(\mathbf{q}^*, \mathbf{p}^*)^T$, or the adjoint variables defined on the domain only.

with

$$
\begin{aligned}
&(\mathbf{q}^*, \boldsymbol{\mathcal{B}}(\Omega, \mathbf{q})) \\
&\quad \equiv \int n^* \left(S_\mathrm{n} - C_n n - \frac{\partial}{\partial\theta} \left(n u_\theta \right) \right) \mathrm{d}\theta \\
&\quad + \int u_{||}^* \left(S_{m u_{||}} - b_\theta \frac{\partial p}{\partial \theta} - \frac{\partial}{\partial\theta} \left(m n u_\theta u_{||} - \eta_\theta^\mathrm{i} \frac{\partial u_{||}}{\partial\theta} \right) \right) \mathrm{d}\theta \\
&\quad + \iint \Phi_\mathrm{n}^* \left(\int \Sigma_\mathrm{t} \Phi_\mathrm{n} C(\theta; \mathbf{v}' \to \mathbf{v}) \mathrm{d}\mathbf{v}' - \mathbf{s} \cdot \mathbf{e}_\theta \frac{\partial \Phi_\mathrm{n}}{\partial\theta} - \Sigma_\mathrm{t} \Phi_\mathrm{n} \right) \mathrm{d}\mathbf{v}\, \mathrm{d}\theta \\
&\quad + n_S^* (n_0 - n)\big|_0 + u_{||,S}^* \left(\frac{\Gamma_{n,0}}{b_\theta n_0} - u_{||} \right)\Bigg|_0 + u_{||,S}^* \left(c_\mathrm{s} - u_{||} \right)\Big|_{L_\theta} \\
&\quad + {\Phi_\mathrm{n}}_{,S}^* \left(-\Phi_{\mathrm{n},+} \right)\big|_0 + {\Phi_\mathrm{n}}_{,S}^* \left(R n u_\theta + \Phi_{\mathrm{n},+} - \Phi_{\mathrm{n},-} \right)\big|_{L_\theta} .
\end{aligned}
$$

Following the theoretical development in Chapter 2, the cost functional $J(\Omega, \mathbf{q}(\Omega))$ is located at a saddle point of the Lagrangian, characterized by the equations

$$
\begin{cases} 0 &= \nabla_{\mathbf{q}^*} L(\Omega, \mathbf{q}, \mathbf{q}^*), \\ 0 &= \nabla_{\mathbf{q}} L(\Omega, \mathbf{q}, \mathbf{q}^*). \end{cases}
$$

The first of these conditions gives the state equations. The second condition is elaborated in the next section, and will lead to the adjoint equations.

Lagrangian using the two-equation kinetic formulation

In the case of a two-equation kinetic formulation (6.28), the integration over velocity space is no longer required. The set of equations (6.24), (6.25), and (6.28) is now merely a set of coupled 1D convection-diffusion equations,

$$
0 = \boldsymbol{\mathcal{B}}_D(\mathbf{q}) = S(\mathbf{q}, \frac{\partial \mathbf{q}}{\partial\theta}) - \frac{\partial}{\partial\theta} \left(C^\theta(\mathbf{q}) - D^\theta(\mathbf{q}) \frac{\partial \mathbf{q}}{\partial\theta} \right), \tag{6.36}
$$

for the state variables $\mathbf{q} = (n, u_{||}, \Phi_{\mathrm{n},+}, \Phi_{\mathrm{n},-})^T$, with

$$
C^\theta(\mathbf{q}) = \begin{pmatrix} n u_\theta \\ m n u_\theta u_{||} \\ \Phi_{\mathrm{n},+} \\ -\Phi_{\mathrm{n},-} \end{pmatrix}, \quad D^\theta(\mathbf{q}) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & \eta_\theta^\mathrm{i} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix},
$$

$$S = \begin{pmatrix} \Sigma_{\mathrm{i}} \left(\Phi_{\mathrm{n},+} + \Phi_{\mathrm{n},-} \right) - C_n n \\ m b_\theta v_{n_{\mathrm{n}},\mathrm{th}} \Sigma_{\mathrm{t}} \left(\Phi_{\mathrm{n},+} - \Phi_{\mathrm{n},-} \right) - m u_{||} \Sigma_{\mathrm{cx}} \left(\Phi_{\mathrm{n},+} + \Phi_{\mathrm{n},-} \right) - b_\theta \frac{\partial p}{\partial \theta} \\ \frac{\Sigma_{\mathrm{cx}}}{2} \left(\Phi_{\mathrm{n},+} + \Phi_{\mathrm{n},-} \right) - \Sigma_{\mathrm{t}} \Phi_{\mathrm{n},+} \\ \frac{\Sigma_{\mathrm{cx}}}{2} \left(\Phi_{\mathrm{n},+} + \Phi_{\mathrm{n},-} \right) - \Sigma_{\mathrm{t}} \Phi_{\mathrm{n},-} \end{pmatrix},$$

and boundary conditions

$$0 = \boldsymbol{\mathcal{C}}_0 = \begin{pmatrix} n_0 - n \\ \frac{\Gamma_{n,0}}{b_\theta n_0} - u_{||} \\ -\Phi_{\mathrm{n},+} \\ 0 \end{pmatrix}, \quad 0 = \boldsymbol{\mathcal{C}}_{\mathrm{L}_\theta} = \begin{pmatrix} 0 \\ c_{\mathrm{s}} - u_{||} \\ 0 \\ R b_\theta u_{||} n + \Phi_{\mathrm{n},+} - \Phi_{\mathrm{n},-} \end{pmatrix}.$$

The Lagrangian is

$$L_D(\Omega, \mathbf{q}, \mathbf{q}^*) = J(\Omega, \mathbf{q}) + \left(\mathbf{q}^*, \boldsymbol{\mathcal{B}}_D(\Omega, \mathbf{q}) \right), \tag{6.37}$$

with

$$\left(\mathbf{q}^*, \boldsymbol{\mathcal{B}}_D(\Omega, \mathbf{q}) \right) \equiv \left(\mathbf{q}^*, \boldsymbol{\mathcal{B}}_D(\mathbf{q}) \right)_\Omega + \mathbf{p}_0^* \cdot \boldsymbol{\mathcal{C}}_0 + \mathbf{p}_{L_\theta}^* \cdot \boldsymbol{\mathcal{C}}_{\mathrm{L}_\theta}, \tag{6.38}$$

and $\mathbf{q}^* = (n^*, u_{||}^*, \Phi_{\mathrm{n},+}^*, \Phi_{\mathrm{n},-}^*)^T$, $\mathbf{p}^* = (n_S^*, u_{||,S}^*, \Phi_{\mathrm{n},+,S}^*, \Phi_{\mathrm{n},-,S}^*)^T$.

6.4 Derivation of the Adjoint Equations

In this section, the formal Lagrangian approach is used to derive the adjoint equations of the 1D model described in Section 6.2. The focus is especially on the derivation of the adjoint kinetic neutral equation, and its interaction with the other adjoint equations.

As before, the adjoint equations are found by setting the derivative of the Lagrangian function with respect to $\mathbf{q}$ equal to zero. Therefore, the model equations are first linearized. Integration by parts then allows to move the differential operators to act on the adjoint variables $\mathbf{q}^*$ instead of on $\delta\mathbf{q}$.

6.4.1 The Linearized Equations

Linearizing the ion continuity and momentum equations with respect to $\mathbf{q}$ gives

$$\frac{\partial}{\partial \theta} \left(\delta n u_\theta + n b_\theta \delta u_{||} \right) = \delta S_{n_{\mathrm{i}}} - C_n \delta n, \tag{6.39}$$

$$\frac{\partial}{\partial \theta} \left(m \left(\delta n u_\theta u_{||} + 2 n u_\theta \delta u_{||} \right) - \eta_\theta^{\mathrm{i}} \frac{\partial \delta u_{||}}{\partial \theta} \right) = \delta S_{m u_{||}} - b_\theta \frac{\partial \delta p}{\partial \theta}. \tag{6.40}$$

In the simple 1D model, there is no linearization of η_θ^{i} since it only depends on T_{i}, which is an input parameter. The influence of the kinetic model is hidden in the linearized source terms. These have to be linearized with respect to the neutral transport flux, but also the detector functions themselves have to be linearized. For the continuity equation, this means

$$\delta S_{n_{\mathrm{i}}} = \int \delta\Phi_{\mathrm{n}}(\theta,\mathbf{v})\Sigma_{\mathrm{i}}\mathrm{d}\mathbf{v} + \int \Phi_{\mathrm{n}}(\theta,\mathbf{v})\delta\Sigma_{\mathrm{i}}\mathrm{d}\mathbf{v}, \tag{6.41}$$

while the linearized momentum source is

$$\begin{aligned}\delta S_{mu_{||}} &= \int \delta\Phi_{\mathrm{n}}(\theta,\mathbf{v})(m(\mathbf{v}\cdot\mathbf{e}_{||})\Sigma_{\mathrm{t}} - mu_{||}\Sigma_{\mathrm{cx}})\mathrm{d}\mathbf{v} \qquad (6.42)\\ &\quad + \int \Phi_{\mathrm{n}}(\theta,\mathbf{v})(m(\mathbf{v}\cdot\mathbf{e}_{||})\delta\Sigma_{\mathrm{t}} - m\delta u_{||}\Sigma_{\mathrm{cx}} - mu_{||}\delta\Sigma_{\mathrm{cx}})\mathrm{d}\mathbf{v}.\end{aligned}$$

Linearization of the kinetic neutral equation proceeds in a similar way,

$$\begin{aligned}\mathbf{s}\cdot\mathbf{e}_\theta\frac{\partial\delta\Phi_{\mathrm{n}}}{\partial\theta} + \Sigma_{\mathrm{t}}\delta\Phi_{\mathrm{n}} + \delta\Sigma_{\mathrm{t}}\Phi_{\mathrm{n}} &= \int \Sigma_{\mathrm{t}}\delta\Phi_{\mathrm{n}}(\theta,\mathbf{v}')C(\theta;\mathbf{v}'\to\mathbf{v})\mathrm{d}\mathbf{v}'\\ &\quad + \int \delta\Sigma_{\mathrm{t}}\Phi_{\mathrm{n}}(\theta,\mathbf{v}')C(\theta;\mathbf{v}'\to\mathbf{v})\mathrm{d}\mathbf{v}' + \int \Sigma_{\mathrm{t}}\Phi_{\mathrm{n}}(\theta,\mathbf{v}')\delta C(\theta;\mathbf{v}'\to\mathbf{v})\mathrm{d}\mathbf{v}'.\end{aligned} \tag{6.43}$$

It is remarked that the collision kernel and cross-sections have to be linearized with respect to the state variables:

$$\delta\Sigma_{\mathrm{t}} = \frac{\partial\Sigma_{\mathrm{t}}}{\partial\mathbf{q}}\delta\mathbf{q}, \quad \delta C = \frac{\partial C}{\partial\mathbf{q}}\delta\mathbf{q}, \quad \ldots$$

This is the equivalent of linearizing for example the neutral pressure diffusion coefficient with respect to $\mathbf{q}$. In a Monte Carlo code such as EIRENE, cross-sections and collision kernels are constructed from fits to numerical or experimental data. In EIRENE, polynomial fits are used. However, the derivatives of a polynomial may be significantly more 'wavy' than the polynomial itself. Therefore, it needs to be examined whether these derivatives can be used in an adjoint code, or if different interpolations are needed.

Linearized boundary conditions

Linearization of the boundary conditions is quite straightforward, and leads to

$$\begin{aligned} &\left\{ \begin{array}{rcl} \delta n &=& 0 \\ n b_\theta \delta u_{||} + \delta n u_\theta &=& 0 \end{array} \right. \qquad \text{at} \quad \theta = 0, \\ &\quad \delta u_{||} \;=\; 0 \qquad\qquad \text{at} \quad \theta = L_\theta, \end{aligned} \tag{6.44}$$

and for the neutral transport equation

$$\left\{ \begin{array}{rcll} \delta \Phi_{\mathrm{n},+} &=& 0 & \text{at} \quad \theta = 0, \\ \delta \Phi_{\mathrm{n},-} &=& R b_\theta u_{||} \delta n + R b_\theta \delta u_{||} n + \delta \Phi_{\mathrm{n},+} & \text{at} \quad \theta = L_\theta. \end{array} \right. \tag{6.45}$$

Linearized Lagrangian

Combining all elements, the Lagrangian linearized with respect to $\mathbf{q}$ for the 1D SOL model considered here is

$$L_{\mathbf{q}} \delta \mathbf{q} = J_{\mathbf{q}} \delta \mathbf{q} + (\mathbf{q}^*, \boldsymbol{\mathcal{B}}_{\mathbf{q}} \delta \mathbf{q}) \tag{6.46}$$

with

$$J_{\mathbf{q}} \delta \mathbf{q} = \, (n u_\theta - \Gamma_{\mathrm{d}})(\, \delta n u_\theta + n b_\theta \delta u_{||})|_{L_\theta}$$

and

$$
\begin{aligned}
&(\mathbf{q}^*, \boldsymbol{\mathcal{B}}_{\mathbf{q}} \delta\mathbf{q}) = \qquad\qquad (6.47)\\
&\int n^* \left(\underbrace{\delta S_{n_\mathrm{i}}}_{(4)} - \underbrace{C_n \delta n - \frac{\partial}{\partial\theta}\left(\delta n u_\theta + n b_\theta \delta u_{||}\right)}_{(1)} \right) \mathrm{d}\theta \\
&+ \int u_{||}^* \left(\underbrace{\delta S_{m u_{||}}}_{(4)} - \underbrace{b_\theta \frac{\partial \delta p}{\partial\theta} - \frac{\partial}{\partial\theta}\left(m u_\theta \left(\delta n u_{||} + 2n\delta u_{||}\right) - \eta_\theta^\mathrm{i} \frac{\partial \delta u_{||}}{\partial\theta}\right)}_{(1)} \right) \mathrm{d}\theta \\
&+ \iint \Phi_\mathrm{n}^*(\theta, \mathbf{v}) \underbrace{\left(\int \Sigma_\mathrm{t} \delta\Phi_\mathrm{n}(\theta, \mathbf{v}') C(\theta; \mathbf{v}' \to \mathbf{v}) \mathrm{d}\mathbf{v}' \right)}_{(3)} \mathrm{d}\mathbf{v}\, \mathrm{d}\theta \\
&+ \iint \Phi_\mathrm{n}^*(\theta, \mathbf{v}) \underbrace{\left(\int \delta\Sigma_\mathrm{t} \Phi_\mathrm{n}(\theta, \mathbf{v}') C(\theta; \mathbf{v}' \to \mathbf{v}) \mathrm{d}\mathbf{v}' \right)}_{(5)} \mathrm{d}\mathbf{v}\, \mathrm{d}\theta \\
&+ \iint \Phi_\mathrm{n}^*(\theta, \mathbf{v}) \underbrace{\left(\int \Sigma_\mathrm{t} \Phi_\mathrm{n}(\theta, \mathbf{v}') \delta C(\theta; \mathbf{v}' \to \mathbf{v}) \mathrm{d}\mathbf{v}' \right)}_{(5)} \mathrm{d}\mathbf{v}\, \mathrm{d}\theta \\
&- \iint \Phi_\mathrm{n}^*(\theta, \mathbf{v}) \left(\underbrace{\mathbf{s} \cdot \mathbf{e}_\theta \frac{\partial}{\partial\theta} (\delta\Phi_\mathrm{n}(\theta, \mathbf{v})) + \Sigma_\mathrm{t} \delta\Phi_\mathrm{n}(\theta, \mathbf{v})}_{(2)} + \underbrace{\delta\Sigma_\mathrm{t} \Phi_\mathrm{n}(\theta, \mathbf{v})}_{(5)} \right) \mathrm{d}\mathbf{v}\, \mathrm{d}\theta \\
&- \left(n_S^* \delta n + u_{||,S}^* \delta u_{||} \right)\Big|_0 - u_{||,S}^* \delta u_{||}\Big|_{L_\theta} \\
&- \Phi_{\mathrm{n},+,S}^* \delta\Phi_{\mathrm{n},+}\Big|_0 + \Phi_{\mathrm{n},-,S}^* (R b_\theta (u_{||} \delta n + \delta u_{||} n) + \delta\Phi_{\mathrm{n},+} - \delta\Phi_{\mathrm{n},-})\Big|_{L_\theta} .
\end{aligned}
$$

The terms labeled (1) are ‘non kinetic’ terms. These are treated in a similar way as in previous chapters. All other terms involve kinetic integrals, and require special attention.

6.4.2 The Adjoint Equations

Now, all terms in the linearized Lagrangian (6.46) involving $\delta\mathbf{q}$ are grouped. To this end, integration by parts is used to move all differential operators to $\mathbf{q}^*$. For the terms (1) in the ion continuity and parallel momentum equations, the results from Chapter 3 can be used,

$$\begin{aligned}
(1) \quad = \quad & \int \delta n \left(u_\theta \frac{\partial n^*}{\partial \theta} + (T_\mathrm{i} + T_\mathrm{e}) \frac{\partial}{\partial \theta}(b_\theta u_{||}^*) + m u_\theta u_{||} \frac{\partial u_{||}^*}{\partial \theta} - C_n n^* \right) \mathrm{d}\theta \\
& + \int \delta u_{||} \left(2mnu_\theta \frac{\partial u_{||}^*}{\partial \theta} + \frac{\partial}{\partial \theta}\left(\eta_\theta^\mathrm{i} \frac{\partial u_{||}^*}{\partial \theta} \right) + nb_\theta \frac{\partial n^*}{\partial \theta} \right) \mathrm{d}\theta \\
& - \left(\delta n u_\theta + nb_\theta \delta u_{||} \right) n^* \Big|_0^{L_\theta} - mu_\theta \left(\delta n u_{||} + 2n\delta u_{||} \right) u_{||}^* \Big|_0^{L_\theta} \\
& - (T_\mathrm{i} + T_\mathrm{e}) \delta n b_\theta u_{||}^* \Big|_0^{L_\theta} - \left(\delta u_{||} \eta_\theta^\mathrm{i} \frac{\partial u_{||}^*}{\partial \theta} - u_{||}^* \eta_\theta^\mathrm{i} \frac{\partial \delta u_{||}}{\partial \theta} \right) \Bigg|_0^{L_\theta} .
\end{aligned}$$

Term (2), which is the 'transport' part of the kinetic equation, can be treated in an analogous way. The velocity coordinate $\mathbf{v}$ is independent of the spatial coordinate θ, so it is not influenced by spatial derivatives:

$$\begin{aligned}
(2) \quad = \quad & \iint \delta\Phi_\mathrm{n}(\theta, \mathbf{v}) \left(\mathbf{s} \cdot \mathbf{e}_\theta \frac{\partial}{\partial \theta}(\Phi_\mathrm{n}^*(\theta, \mathbf{v})) - \Sigma_\mathrm{t} \Phi_\mathrm{n}^*(\theta, \mathbf{v}) \right) \mathrm{d}\mathbf{v}\, \mathrm{d}\theta \\
& - \left(\int \Phi_\mathrm{n}^*(\theta, \mathbf{v}) \mathbf{s} \cdot \mathbf{e}_\theta \delta\Phi_\mathrm{n}(\theta, \mathbf{v}) \mathrm{d}\mathbf{v} \right) \Bigg|_0^{L_\theta} .
\end{aligned}$$

Terms (3) – (5) do not involve spatial derivatives. Integration by parts is therefore not needed here. However, these terms have to be elaborated in order to isolate the variations $\delta\mathbf{q}$. Term (3) will determine the collision kernel for the adjoint equation. Since $\mathbf{v}$ and $\mathbf{v}'$ are independent, the order of integration may be reversed, giving

$$\begin{aligned}
(3) \quad &= \iint \delta\Phi_\mathrm{n}(\theta,\mathbf{v}') \left(\int \Sigma_\mathrm{t} \Phi_\mathrm{n}^*(\theta,\mathbf{v}) C(\theta;\mathbf{v}' \to \mathbf{v}) \mathrm{d}\mathbf{v} \right) \mathrm{d}\mathbf{v}' \, \mathrm{d}\theta \\
&= \iint \delta\Phi_\mathrm{n}(\theta,\mathbf{v}) \left(\int \Sigma_\mathrm{t} \Phi_\mathrm{n}^*(\theta,\mathbf{v}') C(\theta;\mathbf{v} \to \mathbf{v}') \mathrm{d}\mathbf{v}' \right) \mathrm{d}\mathbf{v} \, \mathrm{d}\theta \\
&\equiv \iint \delta\Phi_\mathrm{n}(\theta,\mathbf{v}) \left(\int \Sigma_\mathrm{t} \Phi_\mathrm{n}^*(\theta,\mathbf{v}') C^*(\theta;\mathbf{v}' \to \mathbf{v}) \mathrm{d}\mathbf{v}' \right) \mathrm{d}\mathbf{v} \, \mathrm{d}\theta.
\end{aligned}$$

The second equality follows by renaming the integration variables $\mathbf{v} \to \mathbf{v}'$ and $\mathbf{v}' \to \mathbf{v}$. The last equality defines the adjoint collision kernel as $C^*(\theta;\mathbf{v}' \to \mathbf{v}) \equiv C(\theta;\mathbf{v} \to \mathbf{v}')$. Thus, it is seen that the adjoint process involves *inverse* collisions. In practice, it occurs frequently that the collision kernel depends only on $\mathbf{s} \cdot \mathbf{s}'$. For example, the collision kernel in the 1D model is equivalently expressed as

$$\begin{aligned}
C(\theta;\mathbf{v} \to \mathbf{v}') &= \frac{\Sigma_\mathrm{cx}}{\Sigma_\mathrm{t}} \left(\frac{1}{2}\delta(\mathbf{s}-\mathbf{s}') + \frac{1}{2}\delta(\mathbf{s}+\mathbf{s}') \right) \\
&= \frac{\Sigma_\mathrm{cx}}{\Sigma_\mathrm{t}} \left(\frac{1}{2}\delta(\mathbf{s}\cdot\mathbf{s}'-1) + \frac{1}{2}\delta(\mathbf{s}\cdot\mathbf{s}'+1) \right).
\end{aligned}$$

In this case, $C = C^*$ is self-adjoint.

Terms (4) involve linearization of the plasma source terms with respect to both the neutral transport flux and with respect to the other plasma parameters, see (6.41) and (6.42). The former parts will be referred to as terms $(4)_1$, the latter terms $(4)_2$. Thus,

$$\begin{aligned}
(4)_1 &= \iint \delta\Phi_\mathrm{n}(\theta,\mathbf{v}) \left(n^*\Sigma_\mathrm{i} + m u_{||}^* ((\mathbf{v}\cdot\mathbf{e}_{||})\Sigma_\mathrm{t} - u_{||}\Sigma_\mathrm{cx}) \right) \mathrm{d}\mathbf{v} \, \mathrm{d}\theta, \\
(4)_2 &= \int \delta n \int \Phi_\mathrm{n}(\theta,\mathbf{v}) \left(n^* \frac{\partial \Sigma_\mathrm{i}}{\partial n} + m u_{||}^* \left((\mathbf{v}\cdot\mathbf{e}_{||}) \frac{\partial \Sigma_\mathrm{t}}{\partial n} - u_{||} \frac{\partial \Sigma_\mathrm{cx}}{\partial n} \right) \right) \mathrm{d}\mathbf{v} \, \mathrm{d}\theta \\
&\quad - \int \delta u_{||} \int \Phi_\mathrm{n}(\theta,\mathbf{v}) m \Sigma_\mathrm{cx} \mathrm{d}\mathbf{v} \, \mathrm{d}\theta.
\end{aligned}$$

Finally, the terms labeled (5) in Eq. (6.47) are linearizations of the neutral transport equation with respect to the state of the plasma. In the present model, this means linearization with respect to n, so

$$\begin{aligned}
(5) \quad = \quad & \int \delta n \int \Phi_{\mathrm{n}}^*(\theta,\mathbf{v}) \left(\int \frac{\partial \Sigma_{\mathrm{t}}}{\partial n} \Phi_{\mathrm{n}}(\theta,\mathbf{v}') C(\theta;\mathbf{v}' \to \mathbf{v}) \mathrm{d}\mathbf{v}' \right) \mathrm{d}\mathbf{v}\, \mathrm{d}\theta \\
& + \int \delta n \int \Phi_{\mathrm{n}}^*(\theta,\mathbf{v}) \left(\int \Sigma_{\mathrm{t}} \Phi_{\mathrm{n}}(\theta,\mathbf{v}') \frac{\partial}{\partial n} C(\theta;\mathbf{v}' \to \mathbf{v}) \mathrm{d}\mathbf{v}' \right) \mathrm{d}\mathbf{v}\, \mathrm{d}\theta \\
& - \int \delta n \int \Phi_{\mathrm{n}}^*(\theta,\mathbf{v}) \frac{\partial \Sigma_{\mathrm{t}}}{\partial n} \Phi_{\mathrm{n}}(\theta,\mathbf{v}) \mathrm{d}\mathbf{v}\, \mathrm{d}\theta.
\end{aligned}$$

By grouping terms in (1) – (5), the adjoint equations are found. Terms in δn yield the adjoint continuity equation,

$$\begin{aligned}
-u_\theta \frac{\partial n^*}{\partial \theta} & = (T_{\mathrm{i}} + T_{\mathrm{e}}) \frac{\partial}{\partial \theta} (b_\theta u_{||}^*) + m u_\theta u_{||} \frac{\partial u_{||}^*}{\partial \theta} - C_n n^* \\
& + \int \Phi_{\mathrm{n}}(\theta,\mathbf{v}) \left(n^* \frac{\partial \Sigma_{\mathrm{i}}}{\partial n} + m u_{||}^* \left((\mathbf{v} \cdot \mathbf{e}_{||}) \frac{\partial \Sigma_{\mathrm{t}}}{\partial n} - u_{||} \frac{\partial \Sigma_{\mathrm{cx}}}{\partial n} \right) \right) \mathrm{d}\mathbf{v} \\
& + \int \Phi_{\mathrm{n}}^*(\theta,\mathbf{v}) \left(\int \frac{\partial \Sigma_{\mathrm{t}}}{\partial n} \Phi_{\mathrm{n}}(\theta,\mathbf{v}') C(\theta;\mathbf{v}' \to \mathbf{v}) \mathrm{d}\mathbf{v}' \right) \mathrm{d}\mathbf{v} \\
& + \int \Phi_{\mathrm{n}}^*(\theta,\mathbf{v}) \left(\int \Sigma_{\mathrm{t}} \Phi_{\mathrm{n}}(\theta,\mathbf{v}') \frac{\partial}{\partial n} C(\theta;\mathbf{v}' \to \mathbf{v}) \mathrm{d}\mathbf{v}' \right) \mathrm{d}\mathbf{v} \\
& - \int \Phi_{\mathrm{n}}^*(\theta,\mathbf{v}) \left(\frac{\partial \Sigma_{\mathrm{t}}}{\partial n} \Phi_{\mathrm{n}}(\theta,\mathbf{v}) \right) \mathrm{d}\mathbf{v}. \qquad (6.48)
\end{aligned}$$

The adjoint parallel momentum equation follows by grouping terms in $\delta u_{||}$,

$$- 2mnu_\theta \frac{\partial u_{||}^*}{\partial \theta} - \frac{\partial}{\partial \theta} \left(\eta_\theta^{\mathrm{i}} \frac{\partial u_{||}^*}{\partial \theta} \right) = n b_\theta \frac{\partial n^*}{\partial \theta} - u_{||}^* \int \Phi_{\mathrm{n}}(\theta,\mathbf{v}) m \Sigma_{\mathrm{cx}} \mathrm{d}\mathbf{v}. \qquad (6.49)$$

Finally, the adjoint transport equation for neutrals is

$$\begin{aligned}
& - \mathbf{s} \cdot \mathbf{e}_\theta \frac{\partial}{\partial \theta} (\Phi_{\mathrm{n}}^*(\theta,\mathbf{v})) + \Sigma_{\mathrm{t}} \Phi_{\mathrm{n}}^*(\theta,\mathbf{v}) \ = \qquad (6.50) \\
& \qquad n^* \Sigma_{\mathrm{i}} + m u_{||}^* ((\mathbf{v} \cdot \mathbf{e}_{||}) \Sigma_{\mathrm{t}} - u_{||} \Sigma_{\mathrm{cx}}) + \int \Phi_{\mathrm{n}}^*(\theta,\mathbf{v}') \Sigma_{\mathrm{t}} C^*(\theta;\mathbf{v}' \to \mathbf{v}) \mathrm{d}\mathbf{v}'.
\end{aligned}$$

The boundary conditions are found by grouping all boundary terms at $\theta = 0$ and $\theta = L_\theta$ in the linearized Lagrangian, including the terms from the linearized cost functional. This leads to the boundary conditions for the plasma equations

$$\left\{\begin{array}{lcl} u_{||}^* & = & 0 \\ n_S^* & = & u_\theta n^* \\ u_{||,S}^* & = & n b_\theta n^* + \eta_\theta^{\mathrm{i}} \dfrac{\partial u_{||}^*}{\partial \theta} \end{array}\right. \quad \text{at} \quad \theta = 0,$$
$$\left\{\begin{array}{lcl} n^* & = & (n u_\theta - \Gamma_{\mathrm{d}}) + R\Phi_{\mathrm{n},-}^* \\ u_{||}^* & = & 0 \\ u_{||,S}^* & = & -\eta_\theta^{\mathrm{i}} \dfrac{\partial u_{||}^*}{\partial \theta} \end{array}\right. \quad \text{at} \quad \theta = L_\theta, \tag{6.51}$$

and for the adjoint neutral transport equation,

$$\left\{\begin{array}{lcl} \Phi_{\mathrm{n},-}^* & = & 0 \\ \Phi_{\mathrm{n},+,S}^* & = & \Phi_{\mathrm{n},+}^* \end{array}\right. \quad \text{at} \quad \theta = 0,$$
$$\left\{\begin{array}{lcl} \Phi_{\mathrm{n},+}^* & = & \Phi_{\mathrm{n},-}^* \\ \Phi_{\mathrm{n},-,S}^* & = & \Phi_{\mathrm{n},-}^* \end{array}\right. \quad \text{at} \quad \theta = L_\theta. \tag{6.52}$$

Thus, at $\theta = L_\theta$ there is a boundary condition for the adjoint ion continuity parallel momentum equations (fourth and fifth equation in (6.51)), as well as for the adjoint transport flux $\Phi_{\mathrm{n},+}^*$ (third equation in (6.52)). At $\theta = 0$, there is only a boundary condition for the parallel momentum equation (first equation in (6.51)) and for $\Phi_{\mathrm{n},-}^*$ (first equation in (6.52)), but not for the adjoint ion continuity equation. This is in agreement with the propagation of characteristics in the equations. The convective-diffusive momentum equation needs two boundary conditions, while all other purely convective equations need only boundary conditions at the inlet.

The other equations in (6.51) and (6.52) determine the adjoint multipliers on the boundaries, n_S^*, $u_{||,S}^*$, $\Phi_{\mathrm{n},+,S}^*$ and $\Phi_{\mathrm{n},-,S}^*$. These multipliers relate the adjoint variables to the design equation, i.e. the derivative of the Lagrangian with respect to the control variable. For the 1D model, the design equation will be elaborated in Section 6.5.

Two-equation adjoint kinetic formulation

The derivation of the adjoint equations could be repeated for the case of the two-equation kinetic formulation. However, since decomposing the kinetic equation leads to a set of convection-diffusion equations (in 1D), the result (3.50) with $\boldsymbol{\mathcal{B}}_{\mathbf{q}}^* \mathbf{q}^*$ from (3.45) obtained in Chapter 3 is directly applicable by neglecting all radial terms and setting metric coefficients equal to one. Thus, the adjoint field equations are

$$0 = B^*_{D,\mathbf{q}}\mathbf{q}^* = S^T_{\mathbf{q}}\mathbf{q}^* - \frac{\partial}{\partial\theta}\left(S^T_{\partial_\theta\mathbf{q}}\mathbf{q}^*\right) + \left(C^\theta_{\mathbf{q}}\right)^T\frac{\partial\mathbf{q}^*}{\partial\theta} + \frac{\partial}{\partial\theta}\left((D^\theta)^T\frac{\partial\mathbf{q}^*}{\partial\theta}\right).$$

After elaboration using the definitions of S, C^θ, and D^θ from (6.36), explicit forms are obtained for the adjoint continuity equation,

$$\begin{aligned}-u_\theta\frac{\partial n^*}{\partial\theta} &= (T_\mathrm{i}+T_\mathrm{e})\frac{\partial}{\partial\theta}(b_\theta u^*_{||}) + mu_\theta u_{||}\frac{\partial u^*_{||}}{\partial\theta} - C_n n^* + n^*\frac{\partial\Sigma_\mathrm{i}}{\partial n}\left(\Phi_{\mathrm{n},+}+\Phi_{\mathrm{n},-}\right)\\ &\quad + mu^*_{||}\left(\frac{\partial\Sigma_\mathrm{t}}{\partial n}b_\theta v_{n_\mathrm{n},\mathrm{th}}\left(\Phi_{\mathrm{n},+}-\Phi_{\mathrm{n},-}\right) - \frac{\partial\Sigma_\mathrm{cx}}{\partial n}u_{||}\left(\Phi_{\mathrm{n},+}+\Phi_{\mathrm{n},-}\right)\right)\\ &\quad + \frac{1}{2}\frac{\partial\Sigma_\mathrm{cx}}{\partial n}(\Phi^*_{\mathrm{n},+}+\Phi^*_{\mathrm{n},-})(\Phi_{\mathrm{n},+}+\Phi_{\mathrm{n},-})\\ &\quad - \frac{\partial\Sigma_\mathrm{t}}{\partial n}(\Phi_{\mathrm{n},+}\Phi^*_{\mathrm{n},+}+\Phi_{\mathrm{n},-}\Phi^*_{\mathrm{n},-}),\end{aligned}\tag{6.53}$$

the adjoint momentum equation,

$$-2mnu_\theta\frac{\partial u^*_{||}}{\partial\theta} - \frac{\partial}{\partial\theta}\left(\eta^\mathrm{i}_\theta\frac{\partial u^*_{||}}{\partial\theta}\right) = nb_\theta\frac{\partial n^*}{\partial\theta} - u^*_{||}m\Sigma_\mathrm{cx}\left(\Phi_{\mathrm{n},+}+\Phi_{\mathrm{n},-}\right),\tag{6.54}$$

and the decoupled adjoint transport equations for neutrals,

$$\left\{\begin{array}{rcl} -\frac{\partial\Phi^*_{\mathrm{n},+}}{\partial\theta} + \Sigma_\mathrm{t}\Phi^*_{\mathrm{n},+} &=& S^*_+ + \frac{\Sigma_\mathrm{cx}}{2}(\Phi^*_{\mathrm{n},+}+\Phi^*_{\mathrm{n},-}),\\ \frac{\partial\Phi^*_{\mathrm{n},-}}{\partial\theta} + \Sigma_\mathrm{t}\Phi^*_{\mathrm{n},-} &=& S^*_- + \frac{\Sigma_\mathrm{cx}}{2}(\Phi^*_{\mathrm{n},+}+\Phi^*_{\mathrm{n},-}),\end{array}\right.\tag{6.55}$$

with

$$\left\{\begin{array}{rcl} S^*_+ &=& n^*\Sigma_\mathrm{i} + mu^*_{||}\left(b_\theta v_{n_\mathrm{n},\mathrm{th}}\Sigma_\mathrm{t} - u_{||}\Sigma_\mathrm{cx}\right),\\ S^*_- &=& n^*\Sigma_\mathrm{i} - mu^*_{||}\left(b_\theta v_{n_\mathrm{n},\mathrm{th}}\Sigma_\mathrm{t} + u_{||}\Sigma_\mathrm{cx}\right).\end{array}\right.\tag{6.56}$$

From this last set of equations, it is immediately clear that convective terms have switched sign compared to the forward kinetic equation, while the collision operator is self-adjoint.

The boundary conditions are of course the same as in Eqs. (6.51) and (6.52). This can also verified by elaborating expression (3.52).

6.4.3 Discussion

It is instructive to compare the adjoint equations derived in the previous section to the state equations. As was the case for the convection-diffusion model, there is a large similarity between them. Convective terms have switched sign, and diffusive terms are identical (self-adjoint). Again, similar numerical codes can be used to solve state and adjoint equations. This statement is especially true for the kinetic state and adjoint equations. Here, the typical adjoint equation with inverse collisions and inverse convective velocity is retrieved. Interestingly, adjoint particles $\Phi^*_{\rm n}(\theta, \mathbf{v})$ move in the direction $-\mathbf{v}$. Due to the symmetry of the collision operator, $C^* = C$ and $C(\theta; \mathbf{v}' \to \mathbf{v}) = C(\theta; (-\mathbf{v}') \to (-\mathbf{v}))$ (see above), the adjoint equation (6.50) can be written more conveniently by making the substitutions $\mathbf{v} \to -\mathbf{v}$, $\mathbf{v}' \to -\mathbf{v}'$ and defining $\Psi^*_{\rm n}(\theta, \mathbf{v}) \equiv \Phi^*_{\rm n}(\theta, -\mathbf{v})$, yielding

$$\begin{aligned} \mathbf{s} \cdot \mathbf{e}_\theta \frac{\partial}{\partial \theta}(\Psi^*_{\rm n}(\theta, \mathbf{v})) + \Sigma_{\rm t} \Psi^*_{\rm n}(\theta, \mathbf{v}) \; &= \qquad (6.57) \\ n^* \Sigma_{\rm i} - m u^*_{||}((\mathbf{v} \cdot \mathbf{e}_{||})\Sigma_{\rm t} + u_{||}\Sigma_{\rm cx}) + &\int \Psi^*_{\rm n}(\theta, \mathbf{v}') \Sigma_{\rm t} C(\theta; \mathbf{v}' \to \mathbf{v}) {\rm d}\mathbf{v}', \end{aligned}$$

with boundary conditions from (6.52),

$$\left\{ \begin{array}{rcl} \Psi^*_{{\rm n},+} & = & 0 \\ \Psi^*_{{\rm n},-,S} & = & \Psi^*_{{\rm n},-} \end{array} \right. \quad \text{at} \quad \theta = 0,$$

$$\left\{ \begin{array}{rcl} \Psi^*_{{\rm n},-} & = & \Psi^*_{{\rm n},+} \\ \Psi^*_{{\rm n},+,S} & = & \Psi^*_{{\rm n},+} \end{array} \right. \quad \text{at} \quad \theta = L.$$

Therefore, the adjoint kinetic equation can be solved by using an *identical* Monte Carlo process as in the forward kinetic equation, including the absorption boundary conditions at $\theta = 0$ and the reflection boundary conditions at $\theta = L$. The only change is the source term, which was at the target in the forward problem, and is a distributed source in the adjoint problem. Similar observations hold for the two-equation adjoint kinetic formulation. Making the substitution $\mathbf{v} \to -\mathbf{v}$ now means $\Psi^*_{{\rm n},+} = \Phi^*_{{\rm n},-}$, $\Psi^*_{{\rm n},-} = \Phi^*_{{\rm n},+}$, $S^*_{\Psi+} = S^*_-$, and $S^*_{\Psi-} = S^*_+$, so

$$\left\{ \begin{array}{rcl} \frac{\partial}{\partial \theta}(\Psi^*_{{\rm n},+}) + \Sigma_{\rm t} \Psi^*_{{\rm n},+} & = & S^*_{\Psi+} + \frac{\Sigma_{\rm cx}}{2}(\Psi^*_{{\rm n},+} + \Psi^*_{{\rm n},-}), \\ \\ -\frac{\partial}{\partial \theta}(\Psi^*_{{\rm n},-}) + \Sigma_{\rm t} \Psi^*_{{\rm n},-} & = & S^*_{\Psi-} + \frac{\Sigma_{\rm cx}}{2}(\Psi^*_{{\rm n},+} + \Psi^*_{{\rm n},-}). \end{array} \right. \qquad (6.58)$$

Thus, in order to solve the (decoupled) adjoint kinetic equations, the same forward code with adapted sources and boundary conditions can be used to find $\Psi^*_{{\rm n},+}$ and $\Psi^*_{{\rm n},-}$, followed by the back substitution $\Phi^*_{\rm n}(\theta, \mathbf{v}) = \Psi^*_{\rm n}(\theta, -\mathbf{v})$, or $\Phi^*_{{\rm n},+} = \Psi^*_{{\rm n},-}$ and $\Phi^*_{{\rm n},-} = \Psi^*_{{\rm n},+}$.

The switch between source terms and boundary conditions resembles the typical duality arising in standard *adjoint Monte Carlo* techniques. This standard approach aims at solving the forward Monte Carlo process more efficiently by a dual approach, by noting that

$$\text{solve} \quad T\Phi(\mathbf{r}, \mathbf{v}) = S(\mathbf{r}, \mathbf{v}) \quad \text{to accumulate} \quad \langle g(\mathbf{r}, \mathbf{v}) | \Phi(\mathbf{r}, \mathbf{v}) \rangle$$

is equivalent to

$$\text{solve} \quad T^*\Phi^*(\mathbf{r},\mathbf{v}) = S^*(\mathbf{r},\mathbf{v}) \quad \text{to accumulate} \quad \langle g^*(\mathbf{r},\mathbf{v})|\Phi^*(\mathbf{r},\mathbf{v})\rangle$$

by the choices $S^*(\mathbf{r},\mathbf{v}) = g(\mathbf{r},\mathbf{v})$ and $g^*(\mathbf{r},\mathbf{v}) = S(\mathbf{r},\mathbf{v})$. This can be useful for example if the detector is very small, making it unlikely that particles in the forward Monte Carlo simulation will strike it, while in the adjoint problem the small detector becomes the source [115].

In the case of optimization, the *sources* and *detectors* must be interpreted as *design perturbations* and *design sensitivities*, respectively. See also Refs. [46, 47] for a general interpretation of adjoint problems in (design) optimization. In this context, the design sensitivity quantities which have to be detected by solving the forward kinetic neutral equations are the changes in source terms S_{n_i}, $S_{mu_{||}}$,. . . as they appear in the linearized Lagrangian (6.35) — i.e. linearized with respect to Φ_n, multiplied with appropriate adjoint variables and summed over all (plasma) equations. These become the source terms of the adjoint kinetic equation.

The quantities which are sources in the forward kinetic neutral sensitivity equations are perturbations in the plasma state δn, $\delta u_{||}$,. . . resulting from design perturbations δL_θ. These will alter the neutral transport, and become detector functions for the adjoint kinetic equations.

With these observations, the source terms in the adjoint plasma equations can be grouped as follows:

- **Source terms which are linearizations of the detector functions with respect to plasma parameters.** These depend only on Φ_n and can be evaluated while solving the forward Monte Carlo problem. In practice, this means that for every detector function in the forward problem, also its linearization with respect to the other state variables n,$u_{||}$,. . . must be accumulated. Note that if these linearized terms are available, they could also be used to speed up or stabilize *short-cycle*-like iterations of the plasma solver, where a simplified neutral model is used rather than the full kinetic model to reduce computational time.
- **Source terms due to linearization of the kinetic equation itself**. These are linearizations of collision kernels, cross-sections,. . . with respect to the plasma parameters. All these source terms have detector functions which depend on energy (or the velocity component $\mathbf{v}$), and are thus impossible to store in general. These terms are potentially a bottleneck in the application of adjoint Monte Carlo for design optimization, as they require knowledge of the complete neutral distribution. Alternatively, an approximate set of adjoint equations could be used by either neglecting

> these terms, or approximating the neutral distribution locally by for example a (drifting) Maxwellian. It remains to be verified whether this is accurate enough.

6.5 Design Equation

The last equation defining an optimal solution is the design equation, see (2.36). This equation is found by differentiating the saddle point of the Lagrangian with respect to the shape. To this end, the general expressions for shape derivatives given in Chapter 2, Section 2.3.2 may be used. However, in the 1D problem considered here it is easier to differentiate the Lagrangian (6.35) directly with respect to L_θ. It is noted that the derivatives of the domain integrals are zero, because they involve evaluating the state equations on the boundary $\theta = L_\theta$. Therefore, these terms vanish if the state equations are satisfied, and the shape sensitivity becomes

$$\begin{aligned} \dot{J} \;=\; \nabla_{L_\theta} L \;&=\; \left(\Gamma - \Gamma_{\mathrm{d}}\right) \frac{\partial \Gamma}{\partial \theta}\bigg|_{L_\theta} + \lambda \left(L_\theta - L_0\right) - u^*_{||,S} \frac{\partial u_{||}}{\partial \theta}\bigg|_{L_\theta} \\ &\quad + \Phi^{\;*}_{\mathrm{n},S} \frac{\partial}{\partial \theta} \left(R b_\theta u_{||} n + \Phi_{\mathrm{n},+} - \Phi_{\mathrm{n},-}\right)\bigg|_{L_\theta} . \end{aligned} \tag{6.59}$$

Although trivial in this case, it is remarked that again the shape derivative depends only on boundary data. State variables and normal derivatives are required, also of the kinetic fluxes. For the latter, special care is needed, because the derivative of a stochastic quantity is difficult to evaluate.

Since there are no further constraints on the length of the domain, the optimal length $\hat{L}_\theta$ is determined by

$$\dot{J} = 0$$

if the state and adjoint equations are satisfied.

6.6 Conclusion

In this chapter, shape optimization for coupled fluid-kinetic systems is considered on a theoretical level. The continuous adjoint method is used to derive an adjoint fluid-kinetic system, which in turn allows for the efficient computation of shape sensitivities. The different terms in the adjoint kinetic equations are discussed and analyzed with an implementation in coupled fluid-Monte Carlo codes as B2-EIRENE in mind. In particular, it is pointed out that while the details of neutral distribution function are not needed in the forward problem, these details naturally arise in the adjoint kinetic equations. In practical implementations, these terms will probably have to be approximated using reduced models.

Although the derivation has been done based on a simplified 1D model, the results obtained in this chapter are more general. In the next chapter, the practical implementation of the 1D model will be discussed, as well as important practical issues in the treatment of the statistical noise due to the Monte Carlo code. In Chapter 8, the results will applied to a 2D problem involving radiation transport.

Chapter 7

Convergence and Optimization of Coupled Fluid-Kinetic Systems

The theory on the optimization of fluid-kinetic systems presented in Chapter 6 will be applied in practice. To this end, the 1D model described in the previous chapter is used. The aim is to investigate convergence properties, accuracy and computational cost, and set some guidelines for the implementation of the method in larger code systems as B2-EIRENE.

Some details on the implementation of the 1D model are given in Section 7.1. Then, convergence of the coupled system is investigated in Section 7.2. Special attention is paid to the treatment of statistical noise in the residuals of the fluid plasma and adjoint equations. Finally, Section 7.3 analyzes the resulting shape sensitivities and the one-shot optimization algorithm.

7.1 Implementation Aspects

The 1D model described in Section 6.2 and its adjoint have been implemented in a Matlab code. This section highlights some aspects of the implementation. Since the algorithms for state and adjoint equations are very similar, they will be treated jointly.

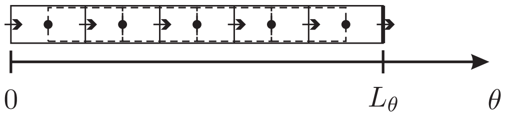


Figure 7.1: Representation of the 1D grid. Collocated cells are indicated with solid lines, staggered momentum cells with dashed lines.

7.1.1 Finite Volume Discretization of the Fluid Equations

In order to discretize the state and adjoint equations, an equidistant grid with n_θ cells of size $d_\theta = L_\theta / n_\theta$ is introduced, see Fig. 7.1. The grid is staggered, with scalar quantities (n, T_i, T_e, T_n, n^*) defined in cell centers and vector quantities ($u_{||}$, $u_{||}^*$) on cell faces. The source terms S_{n_i} and $S_{mu_{||}}$ in the plasma equations are computed in cell centers by the neutral code. All corresponding neutral quantities are also defined in cell centers (f_n, p_n, p_n^*).

A finite volume discretization is used, which is a 1D version of the finite volume discretization discussed in Chapter 3, Section 3.4. For the continuity equation, a first order upwind discretization of the fluxes is used for stability. In the momentum equation, fluxes are obtained with a central scheme.

Also the two-equation kinetic neutral model (6.28) is solved with the finite volume method. Here, a second order upwind scheme is used. Below, this part of the code will also be referred to as the ‘finite volume neutral model’ in order to distinguish it from the Monte Carlo neutral model described in the next paragraph.

In order to solve the coupled nonlinear equations, first the residuals of the equations are computed. Then, as in Section 3.4 an implicit, false-time-stepping procedure is used to iteratively march to the steady-state solution. In order to assess the convergence properties of the coupled fluid-kinetic system, there are options to (1) update all equations sequentially, (2) update the ion continuity momentum equations together and (3) update all equations, including the finite volume neutral model, together. These different relaxation schemes will be further discussed and compared in Section 7.2.1.

7.1.2 Monte Carlo Simulation of the Kinetic Equation

A 1D Monte Carlo code is implemented to simulate the kinetic neutrals and adjoint neutrals. In Section 6.4.3 it has been pointed out that since collision kernel is self-adjoint, the same Monte Carlo process can be used for the forward and adjoint equations (compare equations (6.26) and (6.57)). Only different source sampling routines and detector functions have to be implemented.

Sampling the flight distance can be done based on the distribution function given in Eq. (6.19). A random number ρ_0 is drawn, which determines the distance l_0 the particle will travel. In a finite volume code, all densities and temperatures are assumed constant in a cell. Therefore, the cross-sections are also piecewise constant functions, with value $\Sigma_{\mathrm{t},i}$ in cell i. The integral in Eq. (6.19) can be elaborated as

$$\begin{aligned}\rho_0 &= F_{\mathrm{d}}(l_0) = \int_0^{l_0} \Sigma_{\mathrm{t}}(\mathbf{r}+u\mathbf{s}) \exp\left(-\int_0^u \Sigma_{\mathrm{t}}(\mathbf{r}+t\mathbf{s})\mathrm{d}t\right)\mathrm{d}u \\ &= \sum_{i=1}^{n} \int_0^{\Delta\theta_i} \Sigma_{\mathrm{t},i} \exp\left(-\sum_{j=1}^{i-1} \Sigma_{\mathrm{t},j}\Delta\theta_j \Sigma_{\mathrm{t},i} u\right)\mathrm{d}u \\ &= 1-\prod_{i=1}^{n} \exp\left(-\Sigma_{\mathrm{t},i}\Delta\theta_i\right)\end{aligned}$$

where $\Delta\theta_i$ is the distance traveled in cell i, and $\sum_{i=1}^{n}\Delta\theta_i = l_0$. Thus, the particle continues its flight until $1-\rho_0 = \prod_{i=1}^{n}\exp\left(-\Sigma_{\mathrm{t},i}\Delta\theta_i\right)$.

At the end of the flight, a new random number ρ_1 will determine which collision takes place. In the 1D model, only charge exchange and ionization are possible, so the distribution from which the collision is sampled is very easy. If $\rho_1 < \Sigma_{\mathrm{i}}/\Sigma_{\mathrm{t}}$, the particle ionizes and disappears from the simulation. Otherwise, it experiences a charge-exchange collision. In the latter case, a third random number determines the flight direction after the collision. In the 1D model, the post collision velocity lies with equal probability in the positive and negative θ direction, while the particle speed is not changed. When the new flight direction has been determined, the procedure restarts.

Boundary conditions (6.33) can be implemented easily in the Monte Carlo procedure. When neutrals strike the target, they are reflected. If they reach the core, they are absorbed and therefore removed from the simulation. These conditions hold both for the ‘forward particles’ as for the ‘adjoint particles’.

In practice, particles often ionize quite soon after they are born, before they can penetrate far into the plasma. If no special measures are taken, very little particles would reach the cells close to the core, leading to large standard deviations there. Therefore, in order to ensure that a particle survives long enough, weighted particles are used. Initially, every particle has a weight $w = 1$. In the first stages of its life, a particle is not allowed to ionize. Instead, a charge-exchange collision is forced and the weight of the particle is reduced by the probability of charge exchange (i.e. the weight w is multiplied by the charge-exchange probability $\Sigma_{\mathrm{cx}}/\Sigma_{\mathrm{t}}$). As long as w is larger than some critical value $w_{\min}$, charge exchange is enforced at every collision. Once $w < w_{\min}$, the particle contributes little to the neutral density and source terms, so it is no longer useful to track it. Then, the procedure of weight reduction is stopped, and a random number will again determine whether or not the particle is ionized.

Source sampling

In the forward problem, all neutrals originate at the target due to the recycling boundary condition, see (6.33). Therefore, 'source sampling' is trivial. In the adjoint problem (6.57) the source S^* is distributed over the cells, and can be positive or negative. For source sampling, a distribution function needs to be constructed. This is easily done by noting that the probability that a particle is born in cell j is $\left|S_j^*\right| / \sum_i \left|S_i^*\right|$, where the summation is over all cells. In order to account for the negative particles, the initial weight is either 1 or -1 depending on the sign of the source at its place of birth.

Detector functions

During its flight, the contribution of each particle to the different responses is accumulated with track-length estimators of the form Eq. (6.21). Due to the low dimensionality of the problem, the complete (adjoint) neutral distribution function f_{n} (f_{n}^*) can be computed in each cell. Therefore, each cell i has two detector functions: 1) a detector function

$$g_{i,+}(\mathbf{r},\mathbf{v}) = \chi_i(\mathbf{r})\delta(\mathbf{s}-\mathbf{e}_\theta)v_{n_{\mathrm{n}},\mathrm{th}}^{-1}$$

for particles flying in direction $\mathbf{e}_\theta$, and 2) a detector function

$$g_{i,-}(\mathbf{r},\mathbf{v}) = \chi_i(\mathbf{r})\delta(\mathbf{s}+\mathbf{e}_\theta)v_{n_{\mathrm{n}},\mathrm{th}}^{-1}$$

for particles flying in direction $-\mathbf{e}_\theta$. $\chi_i(\mathbf{r})$ is the characteristic function of cell i. It is 1 inside the cell and 0 outside the cell. The contributions to the track-length estimator are multiplied with the particle weight w to take into account the weight reduction due to forced charge-exchange collisions (see above).

It is remarked that due to the erratic behavior of individual trajectories, each particle may contribute several times to the neutral distribution in a particular cell. When all particles have been simulated, the average value of each estimator is an approximation to the desired response, Eq. (6.22). From the (adjoint) neutral distribution function, all other quantities such as source terms for the fluid equations can be computed.

7.2 Convergence of Coupled Fluid-Kinetic Models

When using a Monte Carlo code to solve for the (adjoint) kinetic neutrals, inevitably statistical noise will be introduced in the fluid equations. This noise also propagates to the fluid quantities as (adjoint) ion density and parallel velocity. As a result, evaluating the accuracy of the simulation becomes more difficult. Furthermore, even assessing the convergence of the fluid code becomes a challenge.

In Sections 7.2.2 and 7.2.3, two coupling schemes between the fluid and kinetic codes will be investigated. Section 7.2.4 will investigate whether a speedup of the code system is possible by eliminating the expensive Monte Carlo code from some of the iterations. However, before including the statistical noise in the coupling of the codes, some interesting conclusions on code speed can already be drawn by considering the two-equation kinetic neutral model, Section 7.2.1.

7.2.1 Iterative Relaxation Schemes

Since the state equations are nonlinear, a solution has to be found iteratively. Therefore, a time-dependent term is introduced in the plasma equations, and implicit correction equations obtained by linearizing the state equations around the current iterate are solved in every iteration to update the ion density and parallel velocity until a steady-state solution is found. Because the kinetic neutral equation is dominated by a very strong ionization sink, no time derivative term is added here in order to avoid strict stability limits on the time step (i.e. in order to avoid negative densities in the linearization step). Thus, the source terms are always updated through the steady-state kinetic neutral equation. The most straightforward iterative scheme, also used for the 2D model described in the first part of this thesis and in edge plasma codes as B2-EIRENE, is a sequential one:

1. Solve the neutral transport equation to obtain the source terms S_{n_i} and $S_{mu_{||}}$ (either with a Monte Carlo or with a finite volume method).
2. Solve the linearized momentum equation to update the parallel velocity (with some additional under relaxation), assuming fixed ion density.

3. Solve the linearized continuity equation to update the ion density (with some additional under relaxation), assuming fixed parallel velocity. Alternatively, use a pressure-correction scheme to update ion density and parallel velocity simultaneously.

Under relaxation is used for stability. Note that in the linearized continuity and momentum equations, also linearization of the source terms with respect to n_i, resp. $u_{||}$, is taken into account. This sequential scheme is compared to two types of coupled solution schemes:

- **Plasma equations coupled** Rather than solving the linearized momentum and ion continuity equations one after the other, a coupled correction of ion density and parallel velocity is computed by linearizing both equations with respect to n_i and $u_{||}$ and solving these linearized equations as one coupled system. It is expected that this method will provide more reliable updates, so larger time steps may be possible and less under relaxation is needed. As in the sequential scheme, source terms S_{n_i} and $S_{mu_{||}}$ are computed at the start of each iteration.
- **Fully coupled** In this last method, all equations are linearized at once, including the kinetic equations, and solved as one big set of equations. At a first glance, this method may appear as a purely academic test to find an ‘upper limit’ on the performance of the coupled iterative solution method. Indeed, the two-equation kinetic formulation has to be used, because it is almost impossible to use such a fully coupled method with a Monte Carlo solver. On the other hand, the fully coupled scheme may also prove useful in the case of *short-cycle* iterations, see Section 7.2.4, where a reduced (fluid) neutral model is used to replace the Monte Carlo solver in some of the iterations.

By introducing more coupling in the iterative scheme, it is expected that a steady-state solution can be found in a smaller number of outer iterations. In case the Monte Carlo solver consumes the bulk of computational time per iteration, a reduction in the number of iterations means a direct reduction in the total simulation time. Furthermore, the coupled system of equations quickly leads to a very large sparse system (especially in multi-fluid simulations), which may open up perspectives for the use of efficient parallelized linear solvers.

In order to compare the different iterative schemes, they are used on a series of test problems. The model parameters are summarized in Table 7.1. Deuterium ions are considered, with mass m twice the proton mass. The viscosity is $\eta_\theta^\mathrm{i} = b_\theta^2 \eta_{||}^\mathrm{i}$, with the parallel viscosity according to Braginskii [16]. The tests are performed for a recycling coefficient varying between 0.5 and 1, because the

Table 7.1: Model parameters.

Parameter	Value	Parameter	Value
L_θ	0.1 m	n_0	$\Gamma_{n,0}/\left(C_n L_\theta + (1-R)c_\mathrm{s} b_\theta\right)$
b_θ	0.1	T_i	40 eV
C_n	$1 \cdot 10^{-4}$ s−1	T_e	40 eV
$\Gamma_{n,0}$	$2 \cdot 10^{23}$ m^{-2} s−1	T_n	40 eV

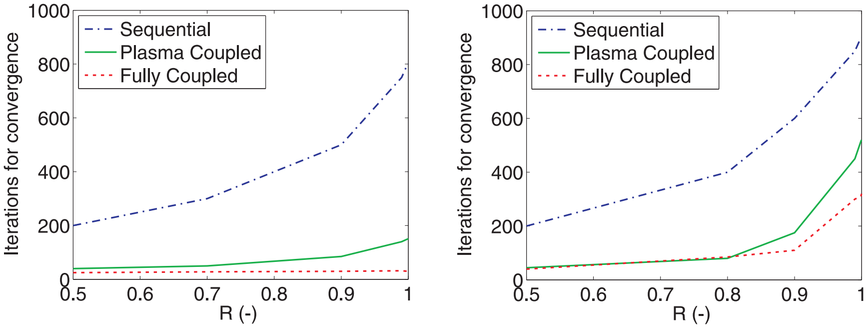
recycling coefficient R has a very large influence on the convergence speed of the code. At low R, there is a strong mass sink at the target, because only part of the ions and electrons reaching the target reenter the domain as neutrals. As $R \to 1$, there is less and less mass leaving the system at the targets. The role of mass sink is now gradually taken over by the loss term $-C_n n$ in the ion continuity equation, which represents radial diffusion. This tends to make the problem stiff. Two types of upstream boundary conditions are considered. In a first set of simulations, the density at the core is allowed to vary with the recycling coefficient, to take into account that the upstream density typically rises with increasing R. In a second set, the upstream density is kept fixed at a rather low value. This makes the problem more stiff, because it reduces the average ion density and therefore the strength of the mass sink due to radial diffusion.

A uniform grid with 1000 cells is used. Since the aim is to get an idea on how well the different coupling schemes handle the nonlinearity in the equations, the finite volume implementation of the kinetic equation is used. Adding the Monte Carlo code would introduce further complexity due to the stochastic noise — this aspect will be treated in Sections 7.2.2 and 7.2.3. For each simulation, the time step was tuned to give convergence in as little as possible iterations. During the scan in recycling coefficient values, the initial state from which a simulation is started is the same for all coupling schemes, and is always taken to be the converged solution at the previous (lower) recycling coefficient. It is remarked that the sequential coupling scheme is very robust regarding the initial state. However, especially at high recycling the coupled schemes greatly benefit from a good initial state. Solutions obtained with the three methods are the same up to machine precision.

The results of the recycling coefficient scan with varying upstream density are summarized in Table 7.2 and Fig. 7.2 (a). The profiles of ion density, neutral density and parallel velocity for three different values of the recycling coefficient are shown in Fig. 7.3. It is seen that as R increases, the time step of non fully coupled schemes has to be reduced and the number of iterations required for convergence sharply increases. There is a point at which this increase stagnates

Table 7.2: Time step Δt (s) and number of iterations N required for convergence.

	Sequential		Plasma equations coupled		Fully coupled	
R	Δt	N	Δt	N	Δt	N
0.50	$1 \cdot 10^{-5}$	200	$2 \cdot 10^{-4}$	40	$1 \cdot 10^{-3}$	25
0.70	$9 \cdot 10^{-6}$	300	$2 \cdot 10^{-4}$	50	$1 \cdot 10^{-3}$	28
0.90	$8 \cdot 10^{-6}$	500	$1 \cdot 10^{-4}$	85	$1 \cdot 10^{-3}$	30
0.99	$7.7 \cdot 10^{-6}$	750	$5 \cdot 10^{-5}$	140	$1 \cdot 10^{-3}$	32
0.999	$7 \cdot 10^{-6}$	800	$5 \cdot 10^{-5}$	150	$1 \cdot 10^{-3}$	30
1.00	$7 \cdot 10^{-6}$	800	$5 \cdot 10^{-5}$	150	$1 \cdot 10^{-3}$	30



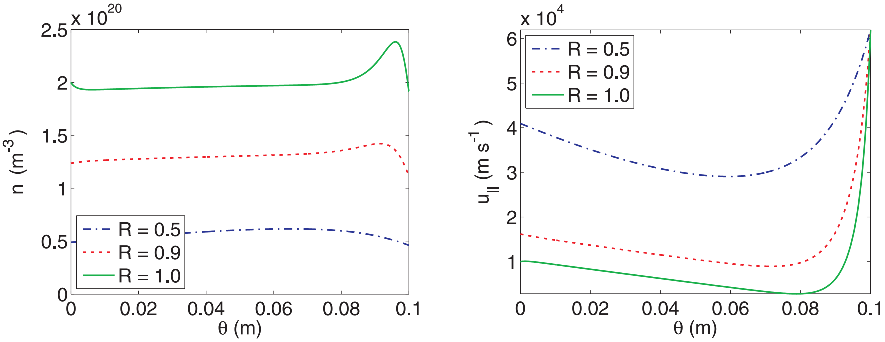
(a)

(b)

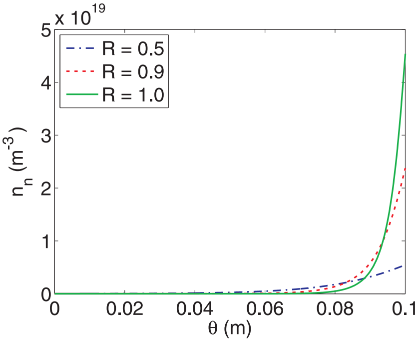
Figure 7.2: Number of iterations required for convergence as a function of recycling coefficient. In figure (a), the core density is increased as R increases (see Table 7.1). In figure (b), core density is the same for all values of R.

($R \approx 0.999$). This is due to the competition between the mass sink at the target (determined by R) and the distributed sink (determined by C_n). Once the latter sink dominates, the influence of R is reduced, and the number of iterations ceases to increase. It appears that the fully coupled scheme suffers less from the increase in recycling coefficient. However, in the second test with fixed upstream density (the value at $R = 0.5$ in the first scan) also the performance of the fully coupled scheme deteriorates, Fig. 7.2 (b). For this model problem, fixing the upstream density means reducing the strength of the ion sink $-C_n n$, since the average density is lower.

These results indicate that depending on the recycling coefficient and the boundary conditions, the coupled system becomes very stiff. Once pumping at the target is no longer the dominant mass sink, the competition between



(a) Ion density (b) Parallel velocity



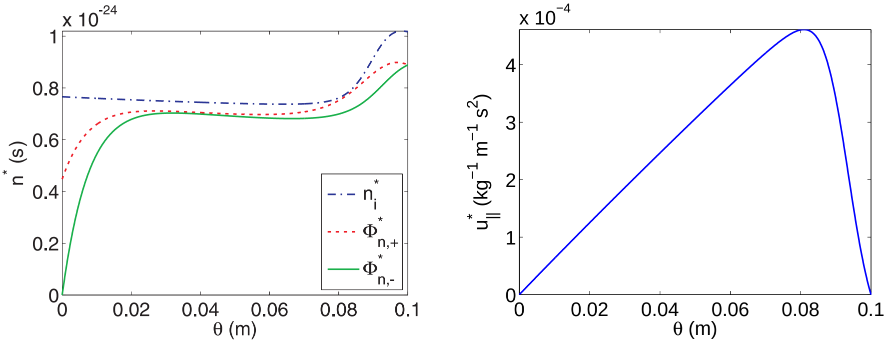
(c) Neutral density

Figure 7.3: Profiles of state variables for different values of R.

fast parallel and slow radial transport comes into play. A mere coupling of all equations can significantly reduce the total number of iterations for convergence in such cases and postpone the performance decrease, but eventually there is sharp rise in iterations (and thus computational time).

The three iterative schemes discussed for the state equations can also be applied to the adjoint equations. Since the adjoint equations are linear, there is an even bigger advantage in solving the fully coupled system. Indeed, in principle the adjoint equations can be solved in exactly one iteration. This can be fully exploited if the finite volume adjoint neutral model is used.

If a Monte Carlo code is used for the kinetic adjoint equations, only the sequential scheme and a partially coupled scheme where the adjoint continuity and momentum equations are solved jointly remain as options. As was the case



(a) (b)

Figure 7.4: (a) Adjoint density and neutral transport flux , and (b) adjoint parallel velocity.

for the state equations, a time-dependent term and under relaxation are added for stability of the iterative scheme.

The complete comparison between the coupling schemes which was performed for the state equations has not been repeated for the adjoint equations. In practice, it is found that the sequential and partially coupled schemes allow to use approximately the same time steps and relaxation factors as for the state equations. As mentioned above, the fully coupled scheme is solved in one iteration, without the need of time stepping or under relaxation.

For the test problems in the remainder of this section, the setup with $R = 0.9$ will be used — a reasonable compromise between nonlinearity and computational speed. In this case, the particle flux at the target is $\Gamma \approx 6.9 \cdot 10^{23}$ m^{-2} s^{-1}. A corresponding optimization problem is defined by aiming to reduce this flux by approximately 10% to $\Gamma_\mathrm{d} = 6.1 \cdot 10^{23}$ m^{-2} s^{-1}. A regularization term is added to the cost functional by choosing $L_0 = 0.1$ m and $\lambda = L_0^{-2} = 100$ m^{-2} in Eq. (6.34). The solution to the adjoint problem in the initial configuration is shown in Fig. 7.4.

7.2.2 Uncorrelated Sampling

When a Monte Carlo code is used to solve the kinetic neutral equation, statistical noise of the Monte Carlo procedure is introduced in the residuals of the fluid equations through the source terms S_{n_i} and $S_{mu_{||}}$. Similarly, the integral terms on the right hand sides of the adjoint fluid equations (6.48) and (6.49) lead

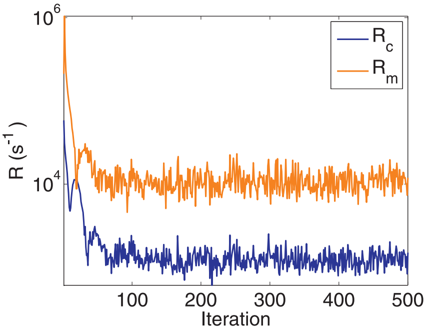


Figure 7.5: Influence of stochastic noise on the residuals of the plasma equations. The residuals fail to decrease due to the use of new random numbers in every iteration.

to noise in these equations stemming from both the kinetic neutral equation and the adjoint kinetic neutral equation. If no special measures are taken, the random numbers used in the Monte Carlo code are different in every iteration, leading to different statistical noise every time the Monte Carlo code is called. Even though theoretically a steady-state solution may exist, the solver can never reach a *numerical steady state*, because the random statistical noise acts as a time-dependent source which prevents the residuals from reaching machine accuracy. This sampling scheme is referred to as *uncorrelated* sampling.

Residuals of a typical state simulation using uncorrelated sampling are illustrated in Fig. 7.5. After a reduction of the residuals in the initial iterations, the residuals then stay at a certain level determined by the noise. The behavior of such residuals has been studied extensively, see for example Ref. [80] and the master's theses of B. Vercammen en K. Ghoos [125, 42]. It is well established that the level at which the residuals oscillate — the *saturation* level — is proportional to $n_\mathrm{t}^{-1/2}$. Although strict convergence to machine precision can only be reached in the limit of an infinite number of particles[1], this does not mean that the solutions obtained with a finite number of particles are useless. However, careful interpretation of the simulation results is mandatory.

Looking at a time trace of the solution, for example Fig. 7.6, it is seen that the solution tends to oscillate around an average value. For a quantity ϕ, this time-averaged value is denoted by $\overline{\phi}$. Intuitively, it is natural to assume that $\overline{\phi}$ approximates the steady-state value ϕ_0. In the case of a linear time invariant system of equations, this can indeed be shown theoretically. For the linear

---

[1] Numerically, the stochastic variations on the sources must be smaller than machine precision in order to achieve convergence.

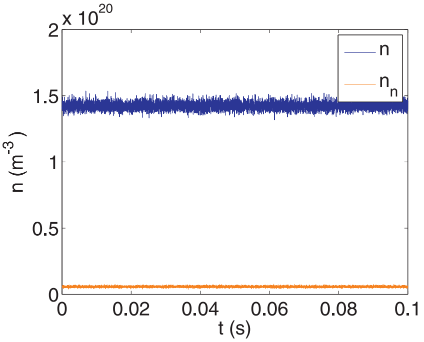


Figure 7.6: Time trace of ion and neutral density in a cell at $\theta = 0.09$ m.

system

$$\frac{\partial \phi}{\partial t} + L(\phi) = S(\phi),$$

assume there is a steady-state solution ϕ_0 which satisfies $L(\phi_0) = S(\phi_0)$, so

$$\frac{\partial}{\partial t}(\phi - \phi_0) + L(\phi - \phi_0) = S(\phi - \phi_0).$$

The stochastic noise from the Monte Carlo code may be modeled as an additional source term σ_ϕ. Due to the central limit theorem, this noise is normally distributed with mean zero and standard deviation proportional to $n_{\mathrm{t}}^{-1/2}$. The modified time-dependent equation which accounts for this noise is then

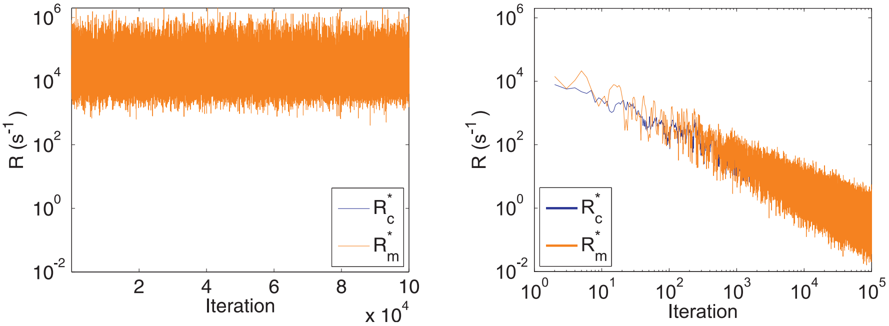
$$\frac{\partial}{\partial t}(\phi - \phi_0) + L(\phi - \phi_0) = S(\phi - \phi_0) + \sigma_\phi.$$

Since the system is linear and time invariant, its output can be computed as a convolution of the source (input) and the impulse response function $h(t)$,

$$\phi(t) - \phi_0 = \int \left(S(\phi(t-\tau) - \phi_0) + \sigma_\phi(t-\tau)\right) h(\tau)\, \mathrm{d}\tau.$$

The first term in the convolution describes the transient phase where the system relaxes from its initial state $\phi(0)$ to the steady-state solution ϕ_0. The second term, which is determined by the stochastic noise, is the problematic term that leads to oscillations. By decomposing $\phi(t)$ as $\phi(t) = \phi_S(t) + \phi_{\sigma_\phi}(t)$ to indicate the responses to the different sources, the effect of the stochastic source may be isolated. Integrating $\phi_{\sigma_\phi}(t) - \phi_0$ over time yields

$$\int \left(\phi_{\sigma_\phi}(t) - \phi_0\right)\, \mathrm{d}t = \iint \sigma_\phi(t-\tau) h(\tau)\, \mathrm{d}\tau\, \mathrm{d}t = \iint \sigma_\phi(t-\tau)\, \mathrm{d}t\, h(\tau)\, \mathrm{d}\tau = 0. \tag{7.1}$$

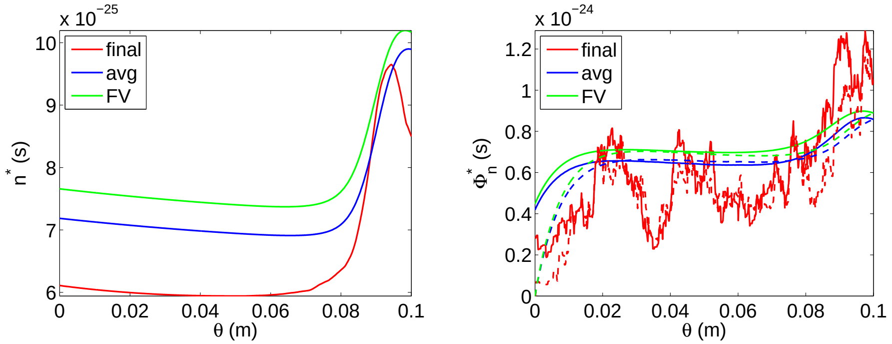


(a) (b)

Figure 7.7: (a) Residuals $R^*_{\mathrm{c}}(\phi^*)$ and $R^*_{\mathrm{m}}(\phi^*)$ of adjoint continuity and momentum equations in case of uncorrelated sampling. (b) $R^*_{\mathrm{c}}(\overline{\phi}^*)$ and $R^*_{\mathrm{m}}(\overline{\phi}^*)$ as a function of iteration number.

Since the noise has an expected value of zero, and assuming its standard deviation is constant, the expected value of the time integral is zero. Therefore, by integrating $\phi(t)$ over time after the initial, transient phase has passed, an approximation $\overline{\phi}$ to the steady-state solution can be obtained. The accuracy of this solution may be assessed by evaluating the residual $R(\overline{\phi}) = S(\overline{\phi}) - L(\overline{\phi})$. This residual should decrease to machine precision as the number of iterations or time steps tends to infinity. This can be illustrated nicely for the adjoint equations, see Fig. 7.7. The left figure shows the residuals of the adjoint equations in a simulation using uncorrelated sampling. In every iteration, only 100 trajectories are simulated by the Monte Carlo code, so the residuals stay at a high level. The right figure shows the residuals of the time-averaged adjoint variables. As the number of iterations increases, these residuals decrease linearly on a double logarithmic scale. Note that this is very slow convergence behavior. Every additional reduction of the residuals by an order of magnitude takes ten times longer than the previous order of magnitude. From the figure, it looks like the oscillations on this average residual tend to increase. This is merely due to the logarithmic scale. Figure 7.8 shows that although the adjoint variables in any particular iteration may be extremely noisy, the average is already quite smooth and tends to the finite volume solution.

For nonlinear systems as the state equations, the averaging result of Eq. (7.1) strictly does not hold. However, since quantities still tend to oscillate around a time average $\overline{\phi}$ (Fig. 7.6), these time averages are worth investigating in more detail. Figure 7.9 compares the ion continuity and momentum residuals $R_{\mathrm{c}}(\phi)$ and $R_{\mathrm{m}}(\phi)$ with the residuals evaluated for the time-averaged solution, $R_{\mathrm{c}}(\overline{\phi})$



(a) (b)

Figure 7.8: (a) Adjoint ion density at final iteration (final), averaged over all iterations (avg) and computed with the finite volume kinetic model (FV). (b) Adjoint neutral transport flux at final iteration (final), averaged over all iterations (avg) and computed with the finite volume kinetic model (FV). Solid lines in (b) represent $\Phi^*_{\mathrm{n},+}$, dashed lines represent $\Phi^*_{\mathrm{n},-}$.

and $R_{\mathrm{m}}(\overline{\phi})$ as a function of iteration number. Again, $n_{\mathrm{t}} = 100$ Monte Carlo particles are used. It is seen that the residuals evaluated for the time-averaged solution are lower than the residuals $R_{\mathrm{c}}(\phi)$ and $R_{\mathrm{m}}(\phi)$. However, after an initial decrease, the nonlinearity of the system prevents $R_{\mathrm{c}}(\overline{\phi})$ and $R_{\mathrm{m}}(\overline{\phi})$ from dropping to machine accuracy as was the case for the adjoint equations. This means that even though the statistical fluctuations can be eliminated by time averaging, these time averages do not satisfy the discretized equations and a 'steady-state' error on the particle and momentum balances remains.

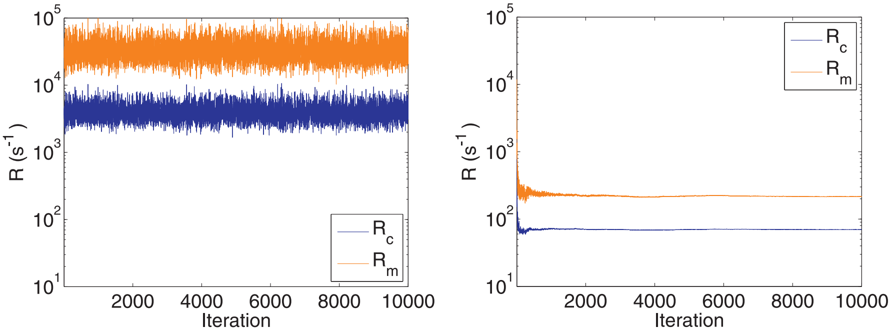
For the 1D model presented here, explicit expressions for $R_{\mathrm{c}}(\overline{\phi})$ and $R_{\mathrm{m}}(\overline{\phi})$ can be obtained. To this end, each quantity ϕ is written as a time average $\overline{\phi}$ plus an error term,

$$\phi = \overline{\phi} + \epsilon_{\phi}.$$

By definition, $\overline{\epsilon_{\phi}} = 0$. The transient equations which are (approximately) solved by the code are of the form

$$\frac{\partial \phi}{\partial t} = R(\phi).$$

Since the time average $\overline{\phi}$ does not change after the initial, transient phase, $\overline{R(\phi)} = 0$. This time-averaged residual can be elaborated for the continuity



(a) (b)

Figure 7.9: (a) Residuals $R_\mathrm{c}(\phi)$ and $R_\mathrm{m}(\phi)$ of continuity and momentum equations in case of uncorrelated sampling. (b) $R_\mathrm{c}(\overline{\phi})$ and $R_\mathrm{m}(\overline{\phi})$ as a function of iteration number.

equation, leading to

$$\begin{aligned} 0 &= \overline{S_\mathrm{n} - C_n n - \frac{\partial}{\partial\theta}\left(n u_\theta\right)} \\ &= \overline{S_\mathrm{n}} - C_n \overline{n} - \frac{\partial}{\partial\theta}\left(\overline{n}\,\overline{u_\theta} + \overline{\epsilon_n \epsilon_{u_\theta}}\right) \\ &= R_\mathrm{c}(\overline{\phi}) - \frac{\partial}{\partial\theta}\left(\overline{\epsilon_n \epsilon_{u_\theta}}\right). \end{aligned} \tag{7.2}$$

Similarly, for the momentum equation the expression

$$0 = R_\mathrm{m}(\overline{\phi}) - \frac{\partial}{\partial\theta}\left(b_\theta \overline{n}\,\overline{\epsilon_{u_{||}}\epsilon_{u_{||}}} + 2\overline{u_\theta}\,\overline{\epsilon_n \epsilon_{u_{||}}} + b_\theta \overline{\epsilon_n \epsilon_{u_{||}} \epsilon_{u_{||}}}\right) + \overline{\epsilon_{S_{mu_{||}}} \epsilon_{u_{||}}} \tag{7.3}$$

is found. Thus, while the levels at which the residuals $R_\mathrm{c}(\phi)$ and $R_\mathrm{m}(\phi)$ saturate scale with $n_\mathrm{t}^{-1/2}$, the residuals $R_\mathrm{c}(\overline{\phi})$ and $R_\mathrm{m}(\overline{\phi})$ are determined by cross correlation terms which scale as n_t^{-1} or $n_\mathrm{t}^{-3/2}$. These dependencies are illustrated in Fig. 7.10.

In order to study the uncorrelated-sampling method, test simulations with 10^2, 10^3 and 10^4 Monte Carlo trajectories per iteration are performed. The time step is 10^{-5} s. It is remarked that the maximum stable time steps allowed in the case of the finite volume neutral model cannot be used reliably with the Monte Carlo solver. For example, referring to Table 7.2, a time step of 10^{-4} s could be used

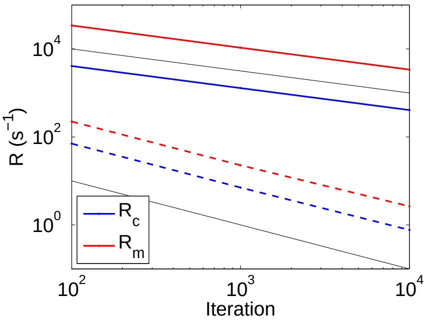
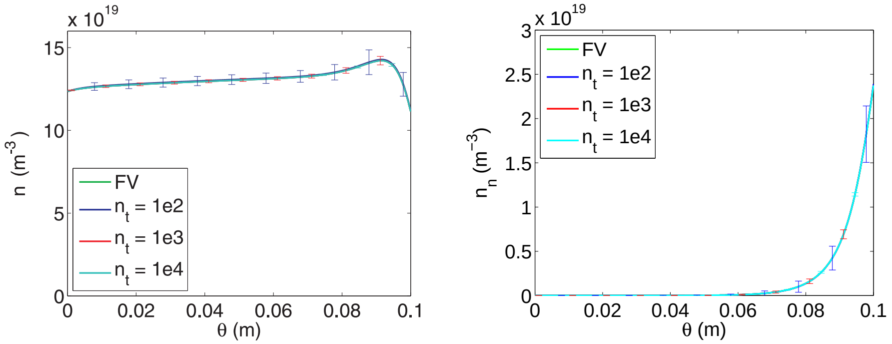


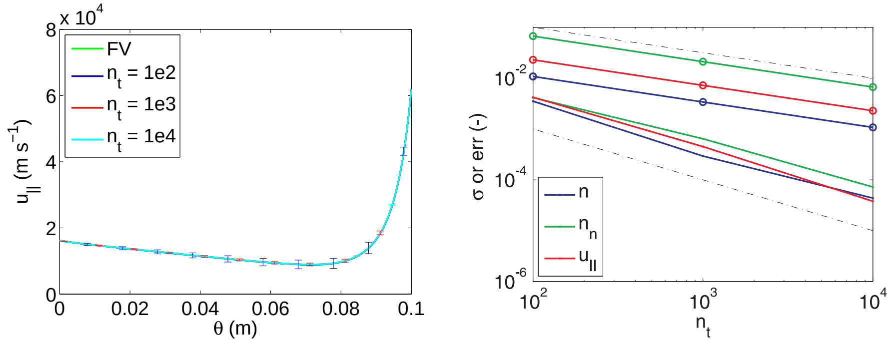
Figure 7.10: Level of the residuals $R_\mathrm{c}(\phi)$ and $R_\mathrm{m}(\phi)$ (solid lines) and $R_\mathrm{c}(\overline{\phi})$ and $R_\mathrm{m}(\overline{\phi})$ (dashed lines) as a function of number of Monte Carlo histories. The black lines represent the n_t^{-1} and $n_\mathrm{t}^{-1/2}$ slopes.

with the partially coupled scheme at $R = 0.9$. With the Monte Carlo code, on the other hand, depending on the specific random numbers or seeds, instabilities may appear at such high time steps. Especially if the number of histories is low, a reduction of the time step by at least an order of magnitude seems necessary for robustness. In Fig. 7.11, the solutions averaged over $n_\mathrm{it} = 10^4$ iterations are shown. For comparison, also the solution obtained with the finite volume neutral solver is provided. Since the finite volume model solves an equation for each direction of flight, it contains only spatial discretization errors. On sufficiently fine grids, as is the case here, these are small. Therefore, the finite volume solution can be seen as a reference solution to benchmark the Monte Carlo results. Error bars in the figure correspond to three times the standard deviation of the time fluctuations on the solution, and can be regarded as a measure for the uncertainty on the result. Figure 7.11 (d) shows that this standard deviation scales as $n_\mathrm{t}^{-1/2}$ as expected. However, assuming individual iterations are statistically independent (which they never truly are) and applying the central limit theorem, the uncertainty on the time-averaged quantities is smaller by a factor of $n_\mathrm{it}^{-1/2}$, i.e. approximately 100 here. Interestingly, the error compared to the finite volume solution, also shown in figure (d), scales as n_t^{-1}.

Concluding, even in the case of nonlinear systems the time average can be taken as an approximation to the steady-state solution. The number of Monte Carlo particles per iteration is important for numerical accuracy. Indeed, both errors and residuals of the average solution — and therefore also local and global particle and momentum balances — scale quite strongly with this number of particles. If the number of particles per iteration is too small, global



(a) Ion density (b) Neutral density



(c) Parallel velocity (d)

Figure 7.11: A comparison of time-averaged profiles obtained for varying n_{t} with finite volume neutral solution. The error bars represent three times the standard deviation of time fluctuations. Figure (d) shows the reduction of the error (solid lines) compared to the finite volume solution as well as the L2-norm of the standard deviation (solid lines with markers).

imbalances may be significant. The errors on the global particle balance for the test problem under consideration are given in Table 7.3. A solution to this problem may be provided by the Robbins-Monro procedure, popular in stochastic approximation [100]. In this method, the noisy sources are integrated over time, and the fluid equations are solved with the averaged sources.

For further reference, it is remarked that the simulation cost C_{US} of the uncorrelated-sampling method is approximately $C_{\mathrm{US}} \sim (n_{\mathrm{it,s}} + n_{\mathrm{it,a}})n_{\mathrm{t}}$, while the (statistical) accuracy scales as $A_{\mathrm{US}} \sim ((n_{\mathrm{it,a}}/n_{\mathrm{it,c}} + 1)n_{\mathrm{t}})^{-1/2}$. $n_{\mathrm{it,s}}$ is the number of iterations required for ‘start up’ — i.e. to reach the steady oscillatory

Table 7.3: Global particle balance as a function of number of Monte Carlo particles.

n_t	Particle imbalance (10^{20} s^{-1})	Relative particle imbalance (%)
$1 \cdot 10^2$	-8.11	-0.4
$1 \cdot 10^3$	-0.79	-0.04
$1 \cdot 10^4$	-0.08	-0.004

state — and can be regarded as a 'fixed cost'. $n_\mathrm{it,a}$ is the number of iterations performed afterwards for time averaging. These iterations allow to remove the statistical fluctuations on the solution. In order to account for possible correlation between iterations during the time-averaging phase, independent samples can only be taken every $n_\mathrm{it,c}$ iterations. It is expected that $n_\mathrm{it,c}$ is related to characteristic time scales of the problem; further research is needed in this respect.

7.2.3 Correlated Sampling

In order to eliminate the time-varying stochastic source, correlated sampling can be used. This technique has been known in the edge plasma community for quite a while, see for example Refs. [41, 80]. In the master thesis of K. Ghoos [42], this method has been studied in detail, leading to the theoretical 'proof of convergence' of this method and extensive numerical validation.

In correlated sampling, each particle has its own fixed *seed* or starting number of the pseudorandom number generator. Therefore, the whole corresponding sequence of random numbers is also fixed, and acts as a deterministic sequence during the iterative procedure. The trajectory of a specific particle is now only determined by the state of the plasma in the cells it crosses, and not by a changing sequence of random numbers in each iteration. This doesn't mean that the entire trajectory of each particle is fixed during the whole simulation. This trajectory may be different between initial and final iterations because the plasma itself is still not in steady state. For example, a particular random number x may lead to a flight distance $L_{\theta,1}$ in iteration one, while the same number x may give a different distance $L_{\theta,2}$ in the next iteration because the plasma has changed. Thus, the next collision of the particle may take place in a different cell compared to the first iteration and so on. However, the trajectory is expected to converge to a specific trajectory, because the plasma properties change less and less. This allows the residuals to reach machine precision, see

Fig. 7.12 (b). The solution obtained from a specific set of seeding numbers is just a *sample solution* and cannot be regarded as the true solution to the complete set of equations. On the other hand, it is a fully consistent solution for a particular instance of frozen noise.

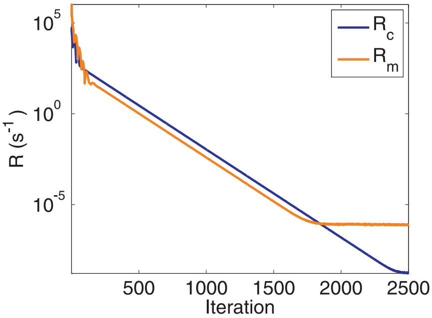


Figure 7.12: In case of correlated sampling, the residuals of the fluid equations go down to machine precision.

In principle, (an approximation to) the solution may be defined as the ensemble average $\langle\phi\rangle \equiv \frac{1}{N}\sum_{i=1}^{N}\phi_i$, where ϕ_i is the solution of a simulation with n_t trajectories obtained with a set of seeds $p_i^{n_\mathrm{t}}$, and the summation is over all possible sets of seeds (theoretically, infinitely many). The question is then whether $\langle\phi\rangle$ is still a solution of the original set of equations, and if not, what the error is.

For linear systems of equations $L(\phi) = S(\phi)$, the ensemble average still satisfies the governing equations. Indeed, since all individual solutions ϕ_i satisfy the linear equation, $L(\phi_i) = S(\phi_i)$, it follows that

$$0 = \langle R(\phi)\rangle = \langle S(\phi)\rangle - \langle L(\phi)\rangle = S(\langle\phi\rangle) - L(\langle\phi\rangle) = R(\langle\phi\rangle). \tag{7.4}$$

In practice, it is not possible to do the simulation for all possible values of $p_i^{n_\mathrm{t}}$, but only for a finite number N. The error or uncertainty on the average solution is determined by the sample standard deviation s_N, defined as

$$s_N = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}\left(\phi_i - \langle\phi\rangle\right)^2}. \tag{7.5}$$

This result is directly applicable to the adjoint equations.

For nonlinear systems of equations, the problem is more complicated, since

$$\langle L(\phi)\rangle \neq L(\langle\phi\rangle) \quad \text{and} \quad \langle S(\phi)\rangle \neq S(\langle\phi\rangle).$$

However, the residual of the ensemble average, $R(\langle\phi\rangle) = S(\langle\phi\rangle) - L(\langle\phi\rangle)$, may be regarded as a measure for the quality of $\langle\phi\rangle$ as a solution. The analysis proceeds in exactly the same way as for the uncorrelated-sampling method discussed in Section 7.2.2. Each quantity ϕ is now written as an ensemble average $\langle\phi\rangle$ plus an error term,

$$\phi = \langle\phi\rangle + \varepsilon_\phi,$$

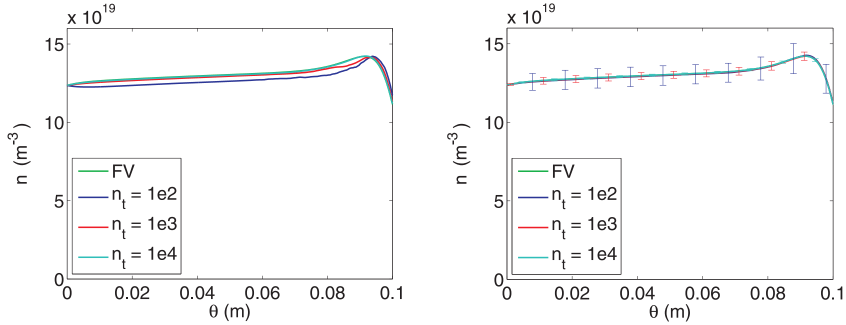
so $\langle\varepsilon_\phi\rangle = 0$. The notation ε_ϕ is used to distinguish this error from the time fluctuation ϵ_ϕ. Plugging this into the continuity equation and averaging this equation over all samples gives

$$\begin{aligned} 0 &= \langle R_\mathrm{c}(\phi)\rangle \\ &= \langle S_\mathrm{n}\rangle - C_n\langle n\rangle - \frac{\partial}{\partial\theta}\left(\langle n\rangle\left\langle u_\theta\right\rangle + \langle\varepsilon_n\varepsilon_{u_\theta}\rangle\right) \\ &= R_\mathrm{c}(\langle\phi\rangle) - \frac{\partial}{\partial\theta}\left(\langle\varepsilon_n\varepsilon_{u_\theta}\rangle\right). \end{aligned} \tag{7.6}$$

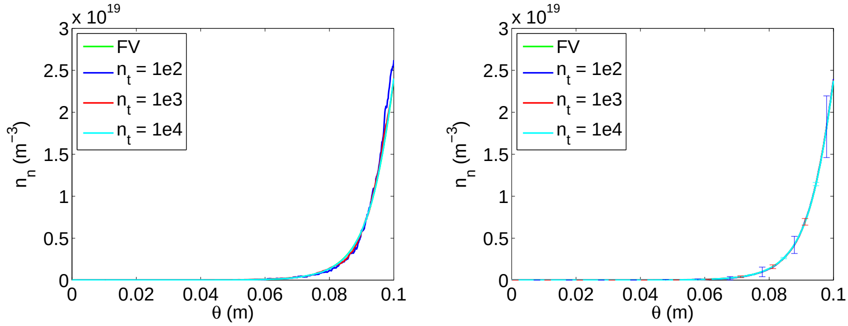
This time, the first equality holds because each individual sample solution satisfies the nonlinear equations. Similarly, for the momentum equation the expression

$$R_\mathrm{m}(\langle\phi\rangle) = \frac{\partial}{\partial\theta}\left(b_\theta\langle n\rangle\left\langle\varepsilon_{u_{||}}\varepsilon_{u_{||}}\right\rangle + 2\langle u_\theta\rangle\left\langle\varepsilon_n\varepsilon_{u_{||}}\right\rangle + b_\theta\langle\varepsilon_n\varepsilon_{u_{||}}\varepsilon_{u_{||}}\rangle\right) - \langle\varepsilon_{S_{mu_{||}}}\varepsilon_{u_{||}}\rangle \tag{7.7}$$

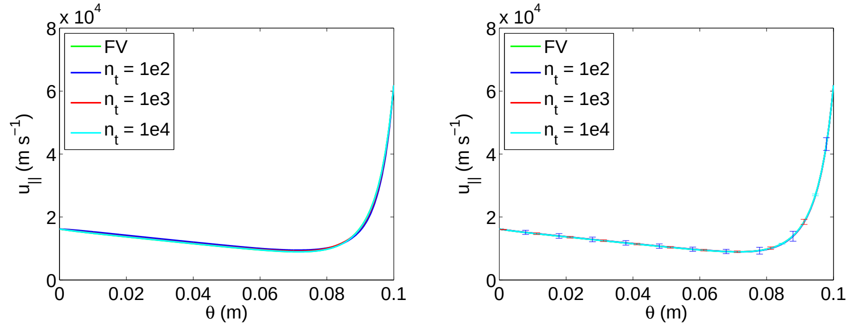
is found. This time, the cross correlation terms which scale as n_t^{-1} or $n_\mathrm{t}^{-\frac{3}{2}}$ can be interpreted as truncation errors. This highlights the analogy with the truncation errors arising due to the approximation of spatial gradients by finite differences in a finite volume code. Due to the finite volume discretization, there is an interplay between spatial and velocity space contributions in these terms. The interesting point is that the truncation errors decrease if $n_\mathrm{t} \to \infty$, showing that the correlated-sampling procedure is consistent.



(a) Ion density (b) Ion density



(c) Neutral density (d) Neutral density



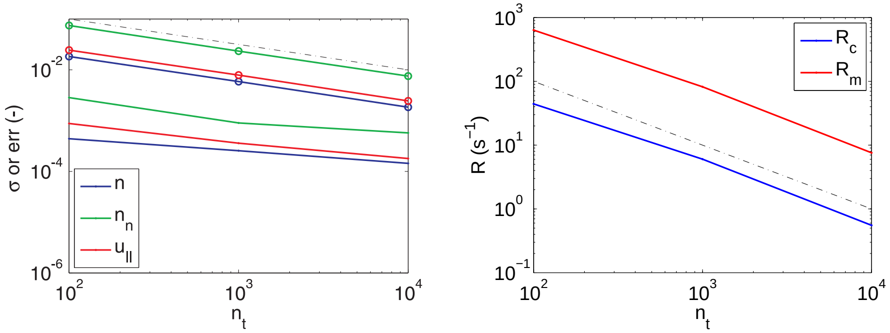
(e) Parallel velocity (f) Parallel velocity

Figure 7.13: A comparison of profiles computed with finite volume and Monte Carlo solvers for the neutrals. On the left, profiles are shown for single runs with 10^2, 10^3 and 10^4 trajectories. The figures on the right compare the corresponding average profiles over 500 samples. The error bars represent three times the local sample standard deviation.

Fig. 7.13 compares the solutions to the state equations obtained with correlated sampling to the solutions obtained with the finite volume neutral model. A solution obtained with correlated sampling is the average over approximately 500 individual simulations with the same number of Monte Carlo trajectories but different seeds. Figures (a), (c) and (e) show density and velocity profiles of one particular sample solution with 10^2, 10^3 and 10^4 trajectories, as well as the finite volume solution. With 10^2 and 10^3 trajectories, there are still pronounced oscillations in the solution due to stochastic noise. For the higher numbers of particles, these are much smaller. In figures (b), (d) and (f), the average density and velocity profiles over all samples with the same number of trajectories are shown. The error bars represent three times the (local) sample standard deviation, i.e. 99% of all individual sample solutions fall within this range. Due to the central limit theorem, the uncertainty on the average profile is smaller by a factor $n^{-1/2}$, with n the number of samples. It is seen that the averaged solutions agree well with the finite volume solution. Only for the case with 10^2 trajectories, the averaged solution can still be distinguished from the finite volume solution. The standard deviation can be interpreted as a (velocity space) discretization error, which decreases as $n_\mathrm{t}^{-1/2}$, see Fig. 7.14 (a). In this figure, also the L2-norm of the error compared to the finite volume solution is shown. In contrast to the uncorrelated-sampling method, this error seems to decrease with $n_\mathrm{t}^{-1/2}$ rather than n_t^{-1}. Finally, the L1-norms of the residuals of the averaged solution are shown in Fig. 7.14 (b). These residuals are the truncation errors described above. As they are determined by covariances, they scale as n_t^{-1}.

Comparing to the results of the uncorrelated-sampling scheme, it is seen that the size of the time fluctuations of the latter scheme roughly corresponds to the size of the statistical fluctuations between samples in the correlated-sampling scheme. It is therefore interesting to compare the methods in terms of computational time required to achieve a certain accuracy. Here, only the accuracy with respect to statistical uncertainty is considered. Increasing the number of trajectories per iteration has the additional effect of reducing the truncation errors and correlation terms in Eqs. (7.2)–(7.3) and (7.6)–(7.7). For the uncorrelated-sampling method, the expressions given at the end of Section 7.2.2 can be used to estimate the accuracy and computational cost. For correlated sampling, the cost and accuracy scale as $C_\mathrm{CS} \sim n\, n_\mathrm{c}\, n_\mathrm{t}$ and $A_\mathrm{CS} \sim (n\, n_\mathrm{t})^{-1/2}$, respectively, where n_c is the number of iterations required for convergence. Typically $n_\mathrm{c} \approx n_\mathrm{it,s}$. In order to have similar statistical accuracy, the requirement is $n \approx n_\mathrm{it,a}/n_\mathrm{it,c} + 1$. Thus, the cost ratio of the correlated-sampling versus uncorrelated-sampling methods is

$$\frac{C_\mathrm{CS}}{C_\mathrm{US}} \sim \frac{n\, n_\mathrm{c}}{(n-1)\, n_\mathrm{it,c} + n_\mathrm{c}}. \tag{7.8}$$



(a) (b)

Figure 7.14: The reduction of (a) the L2-norm of error compared to finite volume solution (solid lines) and sample standard deviation (lines with markers) and (b) the L1-norm of the truncation error. The dashed lines represent the $n_{\mathrm{t}}^{-1/2}$ and n_{t}^{-1} slope, respectively.

Which method is more effective depends on the relative magnitude of the different terms. If $(n-1)\,n_{\mathrm{it,c}} << n_{\mathrm{c}}$ the cost ratio is $C_{\mathrm{CS}}/C_{\mathrm{US}} \approx n$. In this case, uncorrelated sampling would be the method of choice, because its total cost is $(n-1)n_{\mathrm{it,c}} + n_{\mathrm{c}} \approx n_{\mathrm{c}}$, while correlated sampling would require n very expensive simulations. If $(n-1)\,n_{\mathrm{it,c}} >> n_{\mathrm{c}}$ (i.e. if the number of iterations required for convergence can be reduced significantly, or if correlation times for time averaging are long), the cost ratio becomes $C_{\mathrm{CS}}/C_{\mathrm{US}} \approx n_{\mathrm{c}}/n_{\mathrm{it,c}}$. The correlated-sampling procedure may still be more expensive due to the simple fact that full convergence is required for every new sample, while in the uncorrelated method every $n_{\mathrm{it,c}}$ iterations a sample is added. However, if the cost of a simulation can be reduced by a significant factor, e.g. by a factor 10 to 100 using the short-cycle procedure described in the next section, both methods may become comparable in terms of cost.

If $n_{\mathrm{it,c}} = n_{\mathrm{c}}$ or if individual simulations are performed, i.e. if $n = 1$, the cost of correlated and uncorrelated sampling is the same. In the former situation, correlation times are of the order of convergence times. The latter situation occurs if the user is satisfied with the level of statistical uncertainty of a single simulation, or if the statistical fluctuations are of the same order as or smaller than other numerical errors (e.g. spatial discretization errors). Then also grid refinement and using more histories per iteration should be considered to improve the accuracy of the solution, because then spatial discretization errors and the truncation errors and correlation terms in Eqs. (7.2)–(7.3) and

(7.6)–(7.7) are reduced.

An interesting side effect of correlated sampling and the ensemble-averaging procedure is that the global particle and momentum balances evaluated with the averaged fluxes and sources are exact up to machine precision. The finite volume plasma solver combined with source rescaling in the Monte Carlo algorithm ensure perfect balances within each individual simulation. Taking the ensemble average is a linear operation, so this balance is retained. Recalling Table 7.3, this perfect balance cannot be achieved by the time-averaged solution.

Before closing this section, it is remarked that while state simulations using correlated sampling always converge, this is not true for the adjoint equations. In some cases, the residuals fail to decrease to machine precision, but show some oscillatory behavior instead. With a time step of 10^{-5} s, only 25% of the simulations using 10^2 histories converges. When moving to 10^3 and 10^4 histories, approximately 40% and 50% of the simulations converge. The reason why convergence is not achieved for particular sets of seeds is not fully understood, but is probably related to the level of correlation that can be achieved between iterations. In contrast to the state equations where there is only a surface source of particles moving in the negative θ direction at the target, the source in the adjoint kinetic equations is a distributed one. Therefore, more trajectories have to be simulated compared to the forward problem to obtain a similar noise level. For example, on a grid with 1000 cells, using 1000 trajectories in the forward problem turns out to be sufficient to obtain accurate solutions, see Fig. 7.13. In the forward problem these 1000 trajectories all start at one source location, the target, so this source is ‘perfectly sampled’. In the adjoint problem, on the other hand, using only 1000 histories would mean that on average only 1 adjoint particle leaves from each cell. Therefore, at most half of the source is sampled (particles may leave in two directions from each cell!) so there is only a rough approximation of the solution to the adjoint kinetic equation. As a result, the adjoint plasma state (and thus the source of adjoint neutral particles) keeps changing significantly between iterations, and it is much more difficult for individual particles to settle around their final trajectory. This line of thought is further supported by the observation that some adjoint simulations which fail to converge with a time step of 10^{-5} s do converge after a reduction of the time step to 10^{-6} s. In this case, the adjoint state evolves slower. On the downside, this significantly increases the number of iterations needed for convergence.

Also with B2-EIRENE, similar convergence issues are seen with the correlated-sampling procedure. The procedure works well for smaller machines and low recycling conditions [41]. Unfortunately, in larger machines at high recycling conditions it is more difficult if not impossible at present to achieve convergence. Given the potential benefit of using correlated sampling in combination with short cycling, see next section, this issue should be subject of further investigation.

7.2.4 Short Cycling

Very often — especially in this 1D model — a large part of the computational time goes to the solution of the kinetic neutral equations with a Monte Carlo procedure. However, in the initial iterations, the plasma may be far from its final state, so much of this computational time is 'wasted'. Similarly, in the iterations close to convergence, very heavy Monte Carlo runs are performed to give only minor updates in the source terms. In order to reduce computational time, it can be advantageous to replace the detailed but expensive Monte Carlo model by a reduced neutral model in some of the iterations. These iterations will be called *short cycles* following Ref. [80].

An ideal candidate for a reduced neutral model is a (set of) fluid equation(s) derived by taking moments of the kinetic equation (6.4), or its 1D version (6.26). For this simple 1D model, a short-cycle procedure based on the zeroth order moment is used, i.e. the neutral continuity equation

$$\nabla \cdot (n_\mathrm{n} \mathbf{V}_\mathrm{n}) = -n_\mathrm{e} n_\mathrm{n} K_\mathrm{i} \tag{7.9}$$

derived in Eq. (6.16). If volume recombination were included in the 1D model, it would also appear on the right hand side of this equation. Eq. (7.9) can be solved for the neutral density if the neutral velocity is known. Since this short-cycle model is to be used to compute updates in the plasma source terms, which are mainly determined by updates in neutral density, it will be assumed that the higher order moments (in this case, $\mathbf{V}_\mathrm{n}$) do not change, and are known from the last Monte Carlo run[2]. In this way, the closure problem is avoided and the short-cycle model is fully consistent with the last kinetic simulation. The boundary conditions for the short-cycle equation are found by integrating the boundary conditions of the kinetic equation. For the problem under consideration, this means $n_\mathrm{n}\mathbf{V}_\mathrm{n} = -Rb_\theta c_\mathrm{s} n_\mathrm{i}$ at the target.

Eq. (7.9) is now regarded as an equation which relates the neutral, electron, and ion densities around some point $n_{\mathrm{n},0}$, $\mathbf{V}_{\mathrm{n},0}$, $n_{\mathrm{e},0}$, $n_{\mathrm{i},0}$, where subscripts 0 refer to the last solution of the full kinetic equation. This can be made more

---

[2] In the 1D model, the momentum source term also depends on neutral velocity, while the neutral temperature is not needed anymore. Therefore, perhaps a more logical choice would be to still include the neutral momentum equation in the short-cycle model, and assume all second order moments of the distribution function known. However, the neutral continuity equation alone turned out to be sufficient for robust updates.

explicit by writing $n_\mathrm{n} = n_{\mathrm{n},0} + \Delta n_\mathrm{n}$, $n_\mathrm{e} = n_{\mathrm{e},0} + \Delta n_\mathrm{e}$, $n_\mathrm{i} = n_{\mathrm{i},0} + \Delta n_\mathrm{i}$, so

$$\nabla \cdot (\Delta n_\mathrm{n} \mathbf{V}_{\mathrm{n},0}) + n_\mathrm{e} \Delta n_\mathrm{n} K_\mathrm{i} = -\Delta n_\mathrm{e} n_{\mathrm{n},0} K_\mathrm{i} \tag{7.10}$$

with boundary condition $\Delta n_\mathrm{n} \mathbf{V}_{\mathrm{n},0} = -Rb_\theta c_\mathrm{s} \Delta n_\mathrm{i}$ at the target. Indeed, by definition,

$$\nabla \cdot (n_{\mathrm{n},0} \mathbf{V}_{\mathrm{n},0}) = -n_{\mathrm{e},0} n_{\mathrm{n},0} K_\mathrm{i},$$

and $n_{\mathrm{n},0} \mathbf{V}_{\mathrm{n},0} = -Rb_\theta c_\mathrm{s} n_{\mathrm{i},0}$, so these terms cancel from (7.10). Note that (7.10) is a nonlinear approximation to the kinetic equation — the product $\Delta n_\mathrm{e} \Delta n_\mathrm{n}$ is not neglected but is hidden in $n_\mathrm{e} \Delta n_\mathrm{n} K_\mathrm{i} = (n_{\mathrm{e},0} + \Delta n_\mathrm{e}) \Delta n_\mathrm{n} K_\mathrm{i}$. As long as the approximation $\mathbf{V}_{\mathrm{n},0} \approx$ constant holds, the density corrections don't have to be small. Still, the formulation above ensures that if the short-cycle equation (7.10) is solved using the original plasma densities $n_{\mathrm{e},0}$ and $n_{\mathrm{i},0}$, the neutral density $n_{\mathrm{n},0}$ will be retrieved.

Tests using this short-cycle procedure in combination with correlated sampling have been performed. It turns out that the short-cycle model is very robust, and can often be used for many consecutive iterations, with a corresponding reduction in computational time. The typical behavior of the ion continuity and parallel momentum residuals is shown in Fig. 7.15. In this figure, the Monte Carlo code was called every 100 iterations. After every Monte Carlo run, the residuals jump up a bit, but quickly the linear trend returns. Since the neutral continuity equation (7.10) is 're-calibrated' after every Monte Carlo run, the code converges to exactly the same solution as with a Monte Carlo run in every iteration. In Section 7.2.2, it has already been remarked that the maximum stable time steps found in the case of the finite volume neutral model cannot be used reliably for the Monte Carlo solver. For the test problem of Fig. 7.15, a time step of 10^{-6} s was used. This significantly increases the number of iterations required for convergence (from 85 in case of the finite volume neutral model (Table 7.2) to 2500 (Fig. 7.15)). With a time step of 10^{-5} s, the code converges in approximately 500 iterations using correlated sampling. In this case, the short-cycle model could be used for 10 iterations at a time. If the short-cycle model is used longer than an equivalent time step of 10^{-4} s, the code converges around the short-cycle model in between Monte Carlo calls but the total number of iterations needed for convergence increases again. In summary, depending on the case the reduction in computational cost compared to correlated sampling is a factor of 10 to 100.

When using uncorrelated sampling, the short-cycle procedure can only contribute in speeding up the simulation during the initial transient phase. Once the steady oscillatory behavior is reached, new statistical information is needed in order to improve the accuracy of the simulation. This can be achieved most efficiently by using new random numbers in every iteration, so the short-cycle procedure should be switched off.

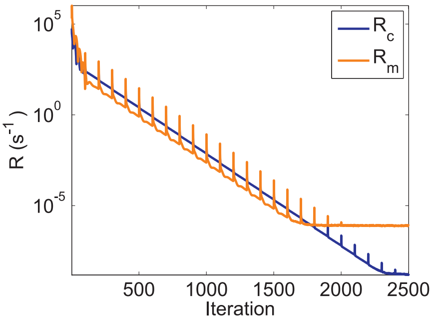


Figure 7.15: Residuals of the plasma equations when short cycling is used with correlated sampling. Intermediate ‘peaks’ in the residuals indicate iterations where a Monte Carlo run is performed.

For more complex edge plasma models, it may be necessary to extend the set of fluid equations for the neutrals by including a momentum equation and perhaps an energy equation. Still, it will remain possible to compute all required higher order moments based on the previous Monte Carlo run, and thus it is expected that such a model will work quite effectively. Looking back at the iterative relaxation schemes discussed in Section 7.2.1, the type of short-cycle model described here would also fit perfectly in a fully coupled iterative scheme. In this way, it may be possible to save even more computational time, but this has not been tested at this point.

In full edge codes as B2-EIRENE, short-cycle schemes have also been implemented. Rather than using a consistent model as in this thesis, these schemes are usually based on rescaling of the plasma sources computed by the Monte Carlo code according to the changing fluxes to the plasma-facing components (i.e. according to the changing neutral particle sources). Again, these schemes work well on smaller machines and at low recycling, but they have difficulties in treating for example volume recombination and are therefore often not used for larger devices at high recycling or detachment. Given the large reduction in computational time, it would be interesting to further investigate the consistent short-cycle models presented in this section for these applications.

As similar reasoning applied to the adjoint equations leads to the short-cycle model

$$-\nabla \cdot \left(n_{\mathrm{n}}^{*}\mathbf{V}_{\mathrm{n}}^{*}\right) = \int S^{*}\,\mathrm{d}\mathbf{v} - n_{\mathrm{n}}^{*} n_{\mathrm{e}} K_{\mathrm{i}}, \tag{7.11}$$

where n_n^* and $\mathbf{V}_\mathrm{n}^*$ are defined by

$$n_\mathrm{n}^* \equiv \int f_\mathrm{n}^* \, \mathrm{d}\mathbf{v}, \qquad n_\mathrm{n}^* \mathbf{V}_\mathrm{n}^* \equiv \int f_\mathrm{n}^* \mathbf{v} \, \mathrm{d}\mathbf{v}. \tag{7.12}$$

Since $\Phi_{\mathrm{n},+}^* = \Phi_{\mathrm{n},-}^*$ at the target, the adjoint neutral velocity $\mathbf{V}_\mathrm{n}^*$ is zero and no boundary condition is required here. At $\theta = 0$, there is an outflow condition. Again, it could be assumed that the higher order moments $(\mathbf{V}_\mathrm{n}^*)$ remain constant during the short-cycle iterations.

Unfortunately, it turns out that the updates computed with this short-cycle model are not reliable — convergence could not be achieved with it. It is not clear why this model is insufficient. Given the large speedup that could be achieved for the state equations, further research effort should be devoted to this issue.

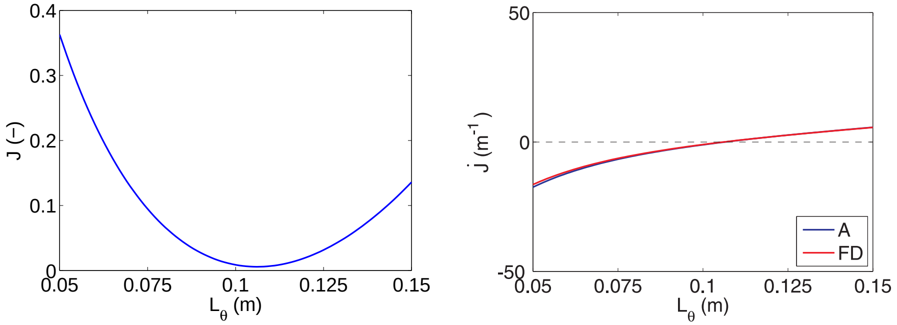
7.3 Particle Flux Optimization

With the tools at hand to solve systems of coupled fluid-kinetic equations in the presence of stochastic noise, the aim is now to optimize such systems using shape sensitivity information. First, the accuracy of the shape sensitivities is investigated in Section 7.3.1. Then, the use of the one-shot scheme for optimization is discussed in Section 7.3.2.

7.3.1 Validation of Sensitivity Computation

The test problem used for the validation of the shape sensitivity is the same as in the previous section. The main model parameters are summarized in Table 7.1. Again, the recycling coefficient is set to $R = 0.9$. The desired particle flux at the target is $\Gamma_\mathrm{d} = 6.1 \cdot 10^{23}$ m^{-2} s^{-1}, i.e. approximately 10% lower than the flux in the initial configuration, $\Gamma \approx 6.9 \cdot 10^{23}$ m^{-2} s^{-1}. A regularization term is added to the cost functional by choosing $L_0 = 0.1$ m and $\lambda = L_0^{-2} = 100$ m^{-2} in Eq. (6.34). With this regularization term, a unique optimum is obtained. However, the regularization term is zero at the initial configuration, so it has no influence on the analysis of the shape sensitivity in this section.

In order to get an idea of the design space, the cost functional is evaluated using the finite volume neutral model for L_θ ranging from 0.05 m to 0.15 m, see Fig. 7.16 (a). The corresponding derivative of the cost functional, computed using both (central) finite differences and the adjoint method, is shown in Fig. 7.16 (b). The relative difference between the derivatives is of the order of a few percent for the grid and parameters chosen for this test.



(a)

(b)

Figure 7.16: (a) Cost functional as a function of domain length and (b) shape sensitivities computed with adjoint (A) and finite difference (FD) methods.

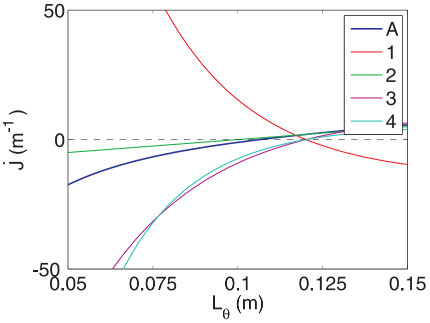


Figure 7.17: Individual components of the shape sensitivity.

It is interesting to compare the magnitude of the individual terms in the shape derivative. To this end, the terms in the shape derivative are labeled (1) to (4) according to their order of appearance on the right hand side of Eq. (6.59):

$$\begin{aligned} \dot{J} &= \underbrace{\left(\Gamma - \Gamma_{\mathrm{d}}\right) \frac{\partial \Gamma}{\partial \theta}\bigg|_{L_\theta}}_{(1)} + \underbrace{\lambda \left(L_\theta - L_0\right)}_{(2)} \underbrace{- \, u^*_{||,S} \frac{\partial u_{||}}{\partial \theta}\bigg|_{L_\theta}}_{(3)} \\ &\quad + \underbrace{\Phi_{\mathrm{n},S}^{\;*} \frac{\partial}{\partial \theta} \left(R b_\theta u_{||} n + \Phi_{\mathrm{n},+} - \Phi_{\mathrm{n},-}\right)\bigg|_{L_\theta}}_{(4)} . \end{aligned} \tag{7.13}$$

These four terms are shown in Fig. 7.17. An important observation is made: the individual terms in the shape sensitivity are quite large, but partially cancel each other when summed.

When using a Monte Carlo procedure to solve the forward and adjoint kinetic equations, different coupling schemes and averaging options can be used at various stages in the process of computing the shape sensitivity. Correlated or uncorrelated sampling can be used in state and adjoint equations. Furthermore, the adjoint equations can be evaluated for a particular sample solution or time instant of the state variables, or for the averaged state solution (i.e. the sample average in case of correlated sampling, and the time average in case of uncorrelated sampling). The two options which are the most relevant to the one-shot scheme are discussed here.

First, correlated sampling is used in state and adjoint equations. In this scheme, the series of random numbers associated with each particle is fixed in both the state and the adjoint equations. Both equations are converged to machine precision one after the other, and then the shape sensitivity is evaluated. By performing a series of runs with different random numbers, different samples are obtained, and the average sensitivity and the statistical variation can be estimated. Table 7.4 shows the average sensitivities and the corresponding variations (three times the standard deviation) for runs with different numbers of histories. Averages are taken over approximately 40 samples. Clearly, if the number of histories is low the shape sensitivity computed from a single run cannot be trusted. Indeed, the uncertainty is larger than the absolute value of the shape sensitivity, so the sign of the shape sensitivity can be wrong. While $n_\mathrm{t} = 10^4$ already leads to good accuracy of the simulation results, this number of histories should clearly still be seen as ‘low’ in view of sensitivity computation. The reason in is the partial cancellation of the different contributions to the shape sensitivity, Fig. 7.17. Although the individual terms in the shape sensitivity are computed with reasonable accuracy, these uncertainties are amplified when the terms are summed. In general, the standard deviation of the sum $\phi_\mathrm{t} = \sum_i \phi_i$ of statistically independent variables ϕ_i, each with standard deviation σ_i, is [115]

$$\sigma_\mathrm{t}^2 = \sum_i \sigma_i^2. \tag{7.14}$$

Therefore, the relative standard deviation $(\sigma_\mathrm{t}^2/\phi_\mathrm{t}^2)^{1/2}$ on the sum will be (significantly) higher than the relative standard deviation on the individual terms $(\sigma_i^2/\phi_i^2)^{1/2}$ if the absolute value of the sum is smaller than that of the individual components. In the case under consideration, the individual terms are not independent, but the general trend still holds. It is noted that given the large uncertainties, even the sign of the *sample average* shape sensitivity can be wrong if the number of samples is small.

Table 7.4: Accuracy of shape sensitivity computation with Monte Carlo neutrals and correlated sampling. The average sensitivity and standard deviation are given for different numbers of histories. Since the second component of the gradient is zero due to the choice of L_0, it is not shown in the table.

	FV	$n_\mathrm{t} = 1 \cdot 10^2$		$n_\mathrm{t} = 1 \cdot 10^3$		$n_\mathrm{t} = 1 \cdot 10^4$	
		Avg.	Std. dev.	Avg.	Std. dev.	Avg.	Std. dev.
$\dot{J}$	-1.08	-0.92	47.96	-1.86	11.98	-1.11	3.66
$\dot{J}_1$	15.37	15.43	4.74	15.37	1.61	15.52	0.42
$\dot{J}_3$	-9.08	-6.03	22.65	-9.26	5.20	-8.45	1.97
$\dot{J}_4$	-7.38	-10.33	35.16	-7.97	9.01	-8.19	2.27

In a second scheme, uncorrelated sampling is used in both state and adjoint equations. The uncorrelated scheme requires new statistical information to be added to both the state and the adjoint equations at every iteration. Therefore, it is proposed to iterate in time by first performing one update of the state equations followed by one update of the adjoint equations, and then compute the sensitivity. New random numbers are used every iteration. After a certain number of iterations, all quantities (state variables, adjoint variables, sensitivities) settle around their time average. They are then averaged over time to remove the statistical fluctuations, while the fluctuations themselves can be taken as a measure for the uncertainty. The result of this procedure is summarized in Table 7.5. Comparison with Table 7.4 shows that for the same number of histories, the size of the time fluctuations is comparable to (somewhat smaller than) the statistical uncertainty in case of correlated sampling. As was the case for correlated sampling, the sign of the time-averaged shape sensitivity can be wrong if the number of time iterations is too small. This is illustrated in Fig. 7.18. For the test problem with $n_\mathrm{t} = 10^2$, the sign of the shape sensitivity did not change anymore after approximately 10 iterations. However, the size of the oscillations suggests that even after 100 iterations the sign could be wrong. After 1000 iterations, the value of the shape sensitivity becomes fairly constant. As the number of trajectories is increased, fewer iterations are required for a reliable gradient.

As has been discussed in Section 7.2.2, the residuals of time-averaged state variables do not drop to machine precision due to the correlation terms. Where the residuals of the adjoint equations did drop to machine precision if the adjoint equations were solved for fixed state variables, this is no longer the case now. Indeed, correlation terms also appear in the adjoint equations since these equations involve products of state and adjoint variables, which both have statistical fluctuations. Figure 7.19 illustrates that residuals of state and

Table 7.5: Accuracy of shape sensitivity computation with Monte Carlo neutrals and uncorrelated sampling. The values of the shape sensitivities are averages over 10000 iterations. The standard deviations are those of the time fluctuations. Since the second component of the gradient is zero due to the choice of L_0, it is not shown in the table.

	FV	$n_{\mathrm{t}} = 1 \cdot 10^2$		$n_{\mathrm{t}} = 1 \cdot 10^3$		$n_{\mathrm{t}} = 1 \cdot 10^4$	
		Avg.	Std. dev.	Avg.	Std. dev.	Avg.	Std. dev.
$\dot{J}$	-1.08	-0.61	31.39	-1.11	9.07	-1.04	2.79
$\dot{J}_1$	15.37	16.38	3.10	15.49	0.96	15.38	0.30
$\dot{J}_3$	-9.08	-9.22	11.25	-9.18	3.41	-9.06	1.04
$\dot{J}_4$	-7.38	-7.77	28.75	-7.42	8.26	-7.36	2.57

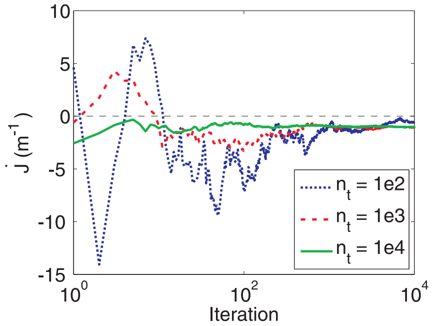
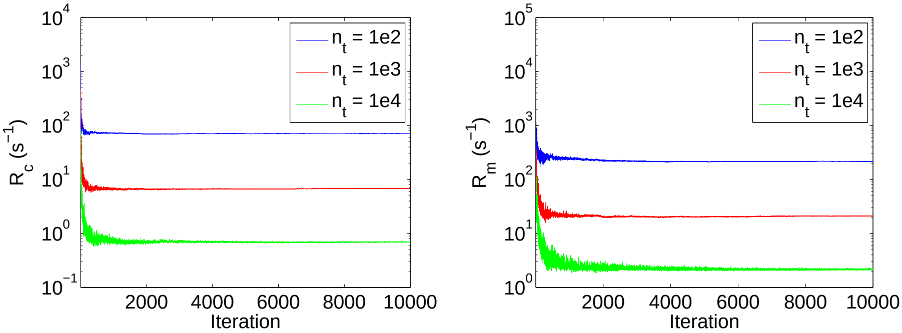


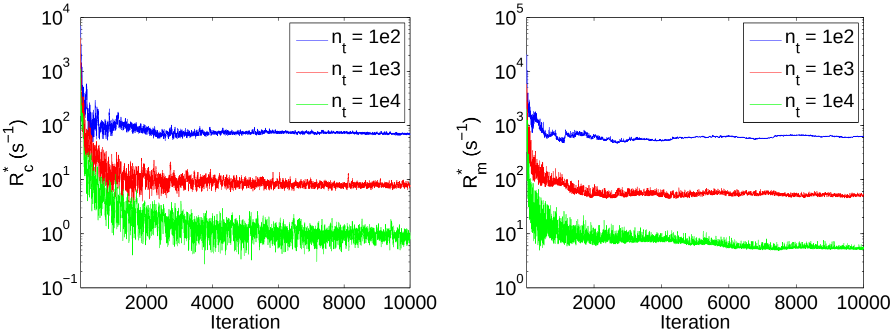
Figure 7.18: Evolution of the time-averaged shape sensitivity for different numbers of trajectories per iteration.

adjoint equations settle around a steady-state value during the computation of the shape sensitivity. Again, the n_{t}^{-1} behavior of the saturation level is clearly seen.

To put the accuracy of the shape sensitivities obtained with the kinetic adjoint neutrals into perspective, the shape sensitivities are also evaluated using the forward Monte Carlo model in a finite difference approach. The computational cost of this procedure is $n_{\mathrm{vars}} n_{\mathrm{c}}$, with n_{vars} the number of design variables and n_{c} the cost of a forward simulation. For the test case, a small domain perturbation of 10^{-6} m is applied to the domain. The statistical noise makes the accurate computation of the sensitivity virtually impossible if different random numbers are used on the perturbed and unperturbed domains, see Table 7.6, second row. The result of Eq. (7.14) is directly applicable — due to the small perturbation the difference in cost functional values is drowned



(a) Continuity equation (b) Momentum equation



(c) Adjoint continuity equation (d) Adjoint momentum equation

Figure 7.19: Evolution of the residuals of the time-averaged solution during sensitivity computation with uncorrelated sampling.

in the noise. However, using correlated sampling a strong correlation can be achieved between the trajectories on the perturbed and unperturbed domains. Therefore, changes in the cost functional are only related to small reactions of individual trajectories to design changes, and not to large differences between trajectories generated by different random numbers. This is one of the most effective methods of computing sensitivities with a Monte Carlo code [115]. The results of this approach are summarized in the last row of Table 7.6. For the same number of histories, the standard deviation of this approach is a factor 100 smaller than that of the adjoint approach. This means that the adjoint method needs approximately 10^4 individual simulations with correlated sampling or 10^4 iterations with uncorrelated sampling in order to achieve the same accuracy. Therefore, the trade-off between forward and adjoint sensitivity analysis will

Table 7.6: Accuracy of Monte Carlo shape sensitivity computation using finite differences, with (CS) and without (FD) correlated sampling. For reference, also the uncertainty on the cost functional is indicated.

	FV	$n_\mathrm{t} = 1 \cdot 10^2$		$n_\mathrm{t} = 1 \cdot 10^3$		$n_\mathrm{t} = 1 \cdot 10^4$	
		Avg.	Std. dev.	Avg.	Std. dev.	Avg.	Std. dev.
J	0.0088	0.0094	0.0038	0.0089	0.0013	0.0004	0.0002
$\dot{J}_\mathrm{FD}$	-1.08	-1.01	$5.1 \cdot 10^3$	-0.98	$2.0 \cdot 10^3$	-0.99	$5.1 \cdot 10^2$
$\dot{J}_\mathrm{CS}$	-1.08	-1.01	0.22	-0.98	0.07	-0.99	0.02

depend on the number of design variables. For the test problem $n_\mathrm{c} \sim 10^2$, so the adjoint method with uncorrelated sampling becomes advantageous if the number of design variables is of the order 10^2 or higher. If the number of design variables is smaller, the forward approach remains the method of choice. Using the adjoint method with correlated sampling seems out of the question. For this 1D problem, it can therefore be recommended to investigate the use of the discrete adjoint approach at the level of the Monte Carlo solver, since this method may combine the advantages of the forward correlated-sampling (low statistical noise) and the backward adjoint (independent of number of design variables) methods. However, this approach would require the complete storage of all Monte Carlo trajectories, since the discrete adjoint method would have to trace them backwards. It remains to be investigated whether or not this is possible in an edge code as B2-EIRENE. Furthermore, in Chapter 8 it will be shown that for the radiation problem, the outcome is exactly opposite. There, the continuous adjoint method has much lower statistical noise compared to the forward approach with correlated sampling.

7.3.2 Solution of the Optimization Problem

With shape sensitivities available, the optimization problem can now be solved. First, the finite volume neutral code is used in order to set a benchmark for the performance of the optimization algorithm with Monte Carlo neutrals. The aim is to investigate the performance of the one-shot scheme for problems with coupled fluid-kinetic systems. Therefore, focus is solely on this scheme. It is remarked that for the 1D problem, a simple (exact) line search in the direction of the control variable would also directly give the optimal solution, but this would not be representative for 2D systems.

The model problem considered in this section is the same as in the previous section. A low recycling ($R = 0.5$) and a high recycling ($R = 0.9$) case are

Table 7.7: Time step Δt (s), gradient relaxation factor r (-) and number of optimization iterations n for fastest convergence of the optimization problem.

R	Δt	r	n
Sequential			
0.50	$1 \cdot 10^{-5}$	$5 \cdot 10^{-4}$	500
0.90	$8 \cdot 10^{-6}$	$4 \cdot 10^{-5}$	2000
Plasma equations coupled			
0.50	$2 \cdot 10^{-4}$	$1 \cdot 10^{-3}$	180
0.90	$1 \cdot 10^{-4}$	$5 \cdot 10^{-5}$	1500
Fully coupled			
0.50	$1 \cdot 10^{-3}$	$2 \cdot 10^{-2}$	100
0.90	$1 \cdot 10^{-3}$	$2.5 \cdot 10^{-3}$	100

considered. All parameters of the high recycling case have been provided above. In the low recycling case, apart from the recycling coefficient also the upstream density is changed according to Table 7.1 and the desired particle flux is reduced to $\Gamma_\mathrm{d} = 2.5 \cdot 10^{23}$ m^{-2} s^{-1}, i.e. again approximately 10% lower than the flux in the initial configuration, $\Gamma \approx 2.9 \cdot 10^{23}$ m^{-2} s^{-1} for $R = 0.5$. The three relaxation schemes discussed in Section 7.2.1 are used in the one-shot algorithm. The design equation is updated by a simple relaxation with constant factor r in the direction of the negative gradient,

$$L_\theta^{n+1} = L_\theta^n - r\dot{J}.$$

Time steps and relaxation factors are optimized to find the solution in as little as possible iterations. The results are summarized in Table 7.7. By comparing the number of iterations needed for the solution of the optimization problem to the corresponding data in Table 7.2, the performance of the optimization algorithm can be deduced.

In all cases, the optimal time step for simulation could also be used in both state and adjoint equations during optimization. With the fully coupled scheme, large steps in design space are allowed and the optimization problem is solved at a total cost of approximately 8 forward simulations for both values of the recycling coefficient. In the sequential scheme, stability requires the time step and relaxation factor to be reduced by roughly the same factor compared to the fully coupled scheme. The total number of iterations required for convergence of the optimization problem is higher, but the overall performance is still very good: optimization is only a factor 5 more expensive than a forward simulation.

The partially coupled scheme is a bit of an exception. At low recycling, the role of the neutrals is not too strong and the decoupling between plasma and neutral equations only leads to a slight decrease in performance compared to the fully coupled scheme. At higher recycling, on the other hand, neutrals become more and more dominant. As was the case when solving the state equations only, the time step for the partially coupled scheme has to be reduced further compared to the fully coupled scheme for stability. However, an even more significant reduction of the relaxation factor in design spaced is required. The cost of optimization becomes significantly higher compared to the cost of a single run, up to a factor of 40 or more. It is interesting to see that at high recycling the number of iterations required with the sequential scheme is not too much higher than with the partially coupled scheme. The reason is that the design step restriction is the limiting factor for the optimization loop, and not the time step. Indeed, in the partially coupled scheme approximately the same convergence speed is achieved if the time step is reduced to 10^{-5} s.

Optimization with the Monte Carlo model requires special care due to the statistical noise on the sensitivities. If the number of trajectories is high, the shape sensitivities obtained with the adjoint method are sufficiently accurate and no additional convergence problems compared to the finite volume model are expected. By extrapolation of Tables 7.4 and 7.5, it is seen that this will be the case if at least $n_\mathrm{t} \approx 10^5$ or higher, because then the statistical variations on the shape sensitivities become of the order of the shape sensitivity or smaller. Theoretically, the method of choice would then be to use correlated sampling for the solution of state and adjoint equations, possibly accelerated by short cycling, since correlated sampling can provide convergence to machine precision. However, working with such a high number of particles involves a significant computational cost, while the accuracy of the simulation was already satisfactory at a lower number of histories. Therefore, the aim is to obtain robust optimization algorithms with a reasonable computational cost, i.e. with a relatively low number of trajectories.

At lower number of particles, the sign of any particular realization of the sensitivity may be wrong. If correlated sampling is used, this can be problematic. Indeed, depending on the seeds the domain could be deformed in the wrong direction. Some example runs for the test problem with $R = 0.9$ are shown in Fig. 7.20. The partially coupled scheme is used. The time step for state and adjoint equations was 10^{-5} s, and the relaxation factor in design space $r = 10^{-5}$. For stability, these values have been reduced compared to the allowed time and design steps given in Table 7.7 for the finite volume neutral model. For some choices of the seeds, a steady-state solution is found. If the number histories is low, sometimes the domain length keeps oscillating around its optimal value or diverges altogether. Diverging runs are not shown in the figure, but depending

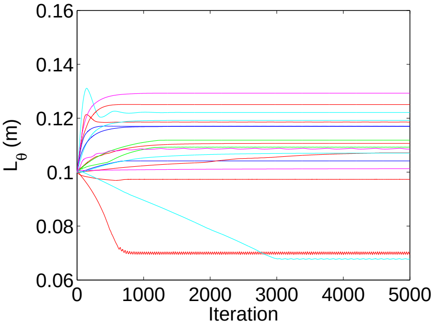
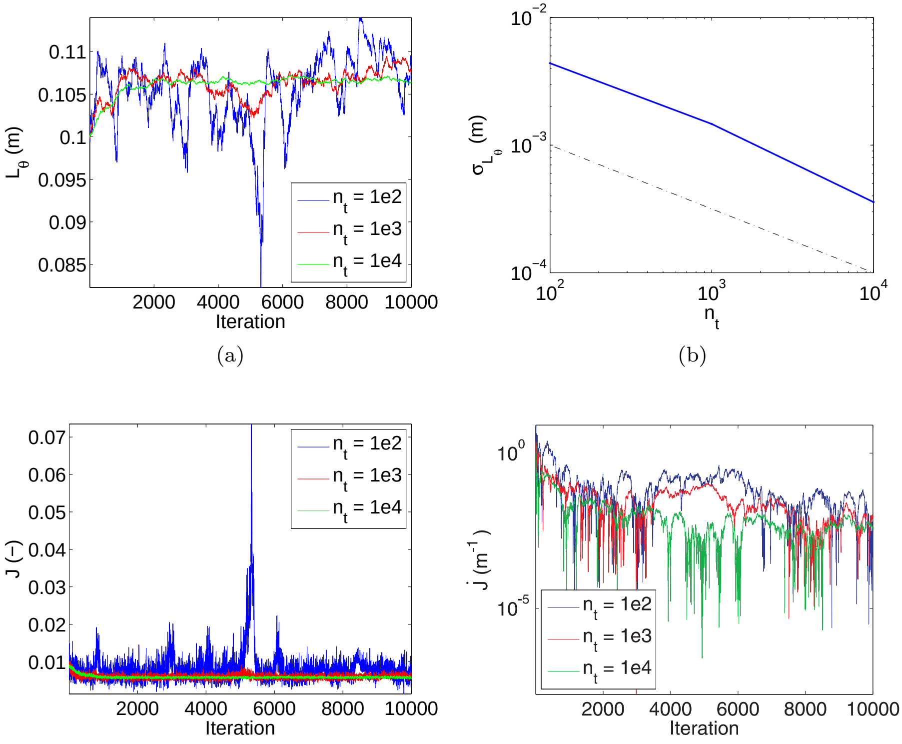


Figure 7.20: Some examples of the evolution of the design variable during one-shot optimization with correlated sampling ($n_{\mathrm{t}} = 10^2$).

on the number of histories over half of the runs can diverge. Of all runs which find a steady-state solution, the average domain length is remarkably close to the optimal value of 0.1065 m. However, it is almost impossible to trust such a result.

When using uncorrelated sampling, the situation is different. Although the accuracy of a simulation with uncorrelated sampling is similar to the accuracy of a simulation with correlated sampling if the number of trajectories per iteration is the same, the uncorrelated scheme converges to the time-averaged solution within each individual simulation. Therefore, it seems to be the more promising candidate for optimization with a low number of histories. The combination with a one-shot scheme is particularly advantageous. Indeed, even though the sign of the sensitivity in a particular iteration may be wrong, the small design steps required in the one-shot scheme act as a smoothing filter guiding the design in the right direction on average over many iterations. Therefore, the statistical variations are much less of an issue. In order to illustrate this, Fig. 7.21 shows the evolution of the domain length during one-shot optimization for different numbers of histories. For these runs, the time step and design relaxation factors were 10^{-5} s and $r = 10^{-5}$, respectively. Once the optimal domain length is reached (after approximately 2500 iterations), oscillations around this optimum will occur. These oscillations can be very high if the number of histories is low. However, the size of the time and design steps can be used to limit the size of the fluctuations to an acceptable level. Fig. 7.22 shows the result of reducing these factor by an order of magnitude for the case with $n_{\mathrm{t}} = 10^3$ histories. The level of the oscillations is reduced to the level of the run with 10^4 histories. However, the figure also shows that more iterations are now needed to settle around the optimal domain length.



(a) (b)

(c) (d)

Figure 7.21: (a) Evolution of design variable during one-shot optimization. (b) Standard deviation of the oscillations on L_θ as a function of number of histories n_t. (c) Value of the the cost functional and (d) evolution of the time-averaged shape sensitivity during optimization.

In order to estimate the computational cost of optimization compared to simulation, a measure for accuracy must be taken into account. Since real convergence cannot be achieved with an uncorrelated-sampling scheme, only qualitative arguments can be made. It is argued that a simulation can be called ‘stationary’ if the fluctuations on the average solution have been averaged out. Referring to Fig. 7.9 (b), this takes approximately 1000 iterations for the problem under consideration. Similarly, an optimal solution is found if the design variable reaches a steady oscillatory state. If n_t is large, 10.000 iterations suffice to be confident about the average optimal domain length. Since these iterations involve state and adjoint equations, the cost of optimization is at least

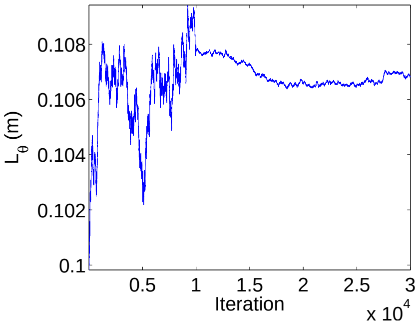


Figure 7.22: History of optimization with $n_\mathrm{t} = 10^3$. After 10.000 iterations, time step and design relaxation factor are reduced by an order of magnitude.

20 times the cost of simulation. If the number of histories per iteration is lower, the average optimal domain length may be harder to identify. Reduction of the relaxation factors may be needed as in Fig. 7.22. This tends to increase the number of iterations for convergence. For example, for $n_\mathrm{t} = 10^3$, one may be confident about the result after at least 30.000 iterations (but possibly more), so the cost is 60 times the cost of simulation. An even further increase in cost is expected with 10^2 histories — this is the price paid for statistical inaccuracy.

7.4 Conclusion

In this chapter, convergence and optimization of coupled fluid-kinetic systems is studied. Special attention is paid to the difficulties introduced by the treatment of the kinetic component with a Monte Carlo code.

First, convergence of coupled fluid-kinetic edge plasma models is studied. It is shown that even without stochastic noise, convergence of such systems becomes very slow at high recycling. In order to mitigate this problem, improved iterative schemes should be considered. For example, by solving the equations in coupled fashion, the number of iterations for convergence — and therefore the total computational cost — can be reduced by a factor of two to three.

Due to stability restrictions, the introduction of statistical noise from the Monte Carlo kinetic model slows down convergence somewhat more. Furthermore, the noise poses challenges on the definition of convergence and the interpretation of the simulated results. The uncorrelated- and correlated-sampling methods are compared. The advantages of the latter include convergence to machine precision, perfect balances at the discrete level and possible acceleration by short cycling.

When computing sensitivities with the continuous adjoint method, the statistical uncertainty can be relatively high. Although on average the computed sensitivity is correct, depending on the number of Monte Carlo histories used even the sign of a particular realization of the sensitivity may be wrong. By averaging over many iterations, the statistical uncertainty can be reduced. For the 1D model problem, the adjoint method outperforms the forward approach using finite differences only if the number of design variables is large. Since the statistical uncertainty on the forward approach is smaller, it can be recommended to investigate the discrete adjoint approach. For the 2D results presented in the next chapter, this result will be exactly opposite.

In the framework of one-shot optimization, the large statistical variation on the shape sensitivity is less problematic. Due to the intrinsically small design steps, the statistical fluctuations are automatically reduced by averaging in ‘design time’. Depending on the number of histories used, the optimization problems can be solved at an equivalent cost of 20 to 100 forward simulations.

Chapter 8

Optimization of Divertor Radiation Load

Especially in large devices as ITER and beyond, a significant fraction of the energy entering the SOL has to be radiated. Although radiation tends to distribute the power uniformly over the main chamber, it also contributes considerably to the divertor target load. This is especially true if a lot of radiation originates in the divertor volume. Therefore, this load should be taken into account in the design.

In this chapter, the aim is to extend the adjoint-based shape optimization algorithms to include radiation. The governing model equations are discussed in Section 8.1. Next, Section 8.2 aims to optimize the total divertor energy load. A cost functional is defined, and the adjoint radiation equations are derived. Since the radiative transfer equations and its adjoint are 5D kinetic equations, they are solved with a Monte Carlo technique. Some aspects of the implementation are discussed in Section 8.3, and the method is validated in Section 8.4. Finally, first results of an optimized divertor are presented in Section 8.5.

8.1 Radiation Transport

Radiation transport is governed by the so-called Radiative Transfer Equation (RTE), which describes the conservation of radiant energy. The equation is written in terms of the radiance or specific intensity $I(\mathbf{r},\mathbf{s},t)$ [66],

$$\frac{1}{c}\frac{\partial I(\mathbf{r},\mathbf{s},t)}{\partial t}+\mathbf{s}\cdot\nabla I(\mathbf{r},\mathbf{s},t)+\mu_{\mathrm{t}}I(\mathbf{r},\mathbf{s},t)=S(\mathbf{r},\mathbf{s},t)+\frac{\mu_{\mathrm{s}}}{4\pi}\int_{4\pi}I(\mathbf{r},\mathbf{s}',t)P(\mathbf{s}'\to\mathbf{s})\,\mathrm{d}\mathbf{s}'. \tag{8.1}$$

This particular form of the equation is for the monochromatic specific intensity, i.e. only one wavelength or frequency is considered. c is the speed of light, μ_{s} the scattering coefficient, μ_{a} the absorption coefficient or opacity, and $\mu_{\mathrm{t}} = \mu_{\mathrm{s}} + \mu_{\mathrm{a}}$ the extinction coefficient. S is the source of radiation (emission), and P is the phase function, which determines the direction in which radiation in scattered. Here and in the remainder of this chapter, it is understood that integration over $\mathrm{d}\mathbf{s}$ (or $\mathrm{d}\mathbf{s}'$) means integration over the differential solid angle $\sin\theta\,\mathrm{d}\theta\,\mathrm{d}\phi$ at $\mathbf{s}$ (or $\mathbf{s}'$), with θ the poloidal and ϕ the azimuthal angle in a local coordinate system. In order to avoid naming conflicts with volume integration below, the usual notation $\boldsymbol{\omega}$ or Ω for the solid angle is avoided. Since the directional component $\mathbf{s}$ involves only 2 dimensions, this is a 6D equation (7D if different wavelengths are taken into account) that can be considered as the radiation transport equivalent of the Boltzmann equation. In general, it is very challenging to solve.

Depending on the application and the accuracy required, various simplifications to the RTE can be made. For example, in order to assess the radiation load of inter-cassette gaps for large tokamaks such as ITER, the geometry can be assumed to be cylindrical due to the large major radius and analytical approximations are sufficient [77]. On the other hand, extensive simulations of the radiation load in detailed ITER divertor geometry have also been performed with commercial ray-tracing software in Ref. [51]. In order to show the full potential of adjoint sensitivity analysis and design optimization, it is chosen to work with a high dimensional model here, however, considerable simplifications to Eq. (8.1) are made.

Only steady-state processes are considered, so the time derivative term cancels, and the time argument is dropped for ease of notation. The source of radiation stems from impurity radiation and radiation from exited states during the ionization of hydrogen. Furthermore, it is assumed that the potential energy of volume recombination is also radiated. The plasma is optically thin for this radiation, so scattering and absorption are neglected. These assumptions lead to the following simplified, yet still 5-dimensional equation:

$$\mathbf{s}\cdot\nabla I(\mathbf{r},\mathbf{s}) = S(\mathbf{r},\mathbf{s}). \tag{8.2}$$

For isotropically emitted radiation, the source is (see also Eq. (3.22))

$$S(\mathbf{r},\mathbf{s}) = S(\mathbf{r}) = \frac{1}{4\pi}\left(E_{\mathrm{i,r}}n_{\mathrm{e}}n_{\mathrm{n}}K_{\mathrm{i}} + E_{\mathrm{r,r}}n_{\mathrm{e}}n_{\mathrm{i}}K_{\mathrm{r}} + c_z n_{\mathrm{i}}n_{\mathrm{e}}L_z\right), \tag{8.3}$$

with $E_{\mathrm{i,r}}$ and $E_{\mathrm{r,r}}$ the energy radiated during ionization and recombination, respectively. Equation (8.2) is linear in the radiance I, but depends nonlinearly on the plasma state variables through the source term. Since absorption of radiation is neglected, the radiation itself has no influence on the plasma, and thus it can be simulated in a post-processing step. For divertor design, the radiation loads of the different components are of interest. In particular, this thesis focuses on the targets. Therefore, it is useful to introduce the energy flux due to radiation, which is obtained by integration over the directional component:

$$\mathbf{J} = \int_{4\pi} I(\mathbf{r}, \mathbf{s})\mathbf{s}\,\mathrm{d}\mathbf{s}. \tag{8.4}$$

Due to the monochromatic assumption, this flux is directly proportional to the photon flux. When looking at a particular surface, the irradiance or power density incident on the surface is found by projecting the energy flux onto the surface, and integrating only over those directions which reach the surface,

$$E = \int_{\mathbf{s}\cdot\boldsymbol{\nu}>0} I(\mathbf{r}, \mathbf{s})\mathbf{s}\cdot\boldsymbol{\nu}\,\mathrm{d}\mathbf{s}. \tag{8.5}$$

As before, $\boldsymbol{\nu}$ is the outward pointing unit normal to the surface. Depending on the properties of the surface, part of this radiation will be absorbed, and part of it will be reflected. This is described by the boundary conditions.

Boundary conditions

As for any partial differential equation, appropriate boundary conditions are needed in order to solve Eq. (8.2). Only the radiance for directions going into the domain can be specified, i.e. directions $\mathbf{s}$ for which $\mathbf{s}\cdot\boldsymbol{\nu} < 0$.

In this thesis, reflection boundary conditions are considered at all surfaces in contact with the plasma. In general, reflection can be specified by the Bidirectional Reflectance Distribution Function (BRDF) $f_r(\mathbf{s}_\mathrm{i}, \mathbf{s}_\mathrm{r})$, which specifies in detail the reflection at a particular surface. $\mathbf{s}_\mathrm{i}$ is the direction of incident radiation and $\mathbf{s}_\mathrm{r}$ that of reflected radiation. Here, a combination of specular and diffuse reflection is assumed.

Perfect specular reflection assumes an ideal flat surface, so that the angles of incidence and reflection are the same. Geometric arguments quickly give the direction of the reflected radiation based on the direction of the incident radiation and the surface normal (see Fig. 8.1 (a)):

$$\mathbf{s}_\mathrm{r} = \mathbf{s}_\mathrm{i} - 2(\mathbf{s}_\mathrm{i}\cdot\boldsymbol{\nu})\boldsymbol{\nu}. \tag{8.6}$$

The inverse relation is also found from geometric arguments, or by noting that $I - 2\boldsymbol{\nu}\boldsymbol{\nu}$ is its own inverse:

$$\mathbf{s}_\mathrm{i} = \mathbf{s}_\mathrm{r} - 2(\mathbf{s}_\mathrm{r}\cdot\boldsymbol{\nu})\boldsymbol{\nu}. \tag{8.7}$$

In terms of radiance of the reflected radiation I_s, specular reflection means

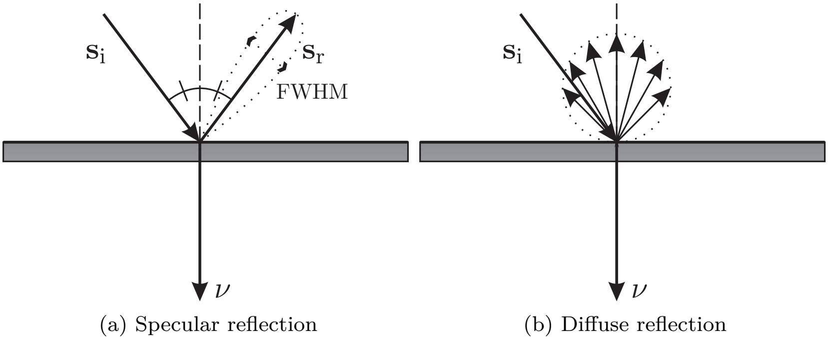
$$I_\mathrm{s}(\mathbf{r},\mathbf{s}) = I(\mathbf{r},\mathbf{s} - 2(\mathbf{s}\cdot\boldsymbol{\nu})\boldsymbol{\nu}), \quad \text{for} \quad \mathbf{s}\cdot\boldsymbol{\nu} < 0. \tag{8.8}$$

Imperfect specular reflection can be approximated by specifying a Gaussian distribution with an angle FWHM (Full Width at Half Maximum) around $\mathbf{s}_\mathrm{r}$, see for example Ref. [4]. This is not taken into account here.

In the case of diffuse reflection, there is no preferred direction for the reflected radiation. This means the radiance I_d of diffusely reflected radiation is uniform:

$$I_\mathrm{d}(\mathbf{r},\mathbf{s}) = I_\mathrm{d}(\mathbf{r}) = \frac{1}{\pi}\int_{\mathbf{s}'\cdot\boldsymbol{\nu}>0} I(\mathbf{r},\mathbf{s}')\mathbf{s}'\cdot\boldsymbol{\nu}\,\mathrm{d}\mathbf{s}', \quad \text{for} \quad \mathbf{s}\cdot\boldsymbol{\nu} < 0. \tag{8.9}$$

This implies that the energy flux from the surface, $I_\mathrm{d}(\mathbf{r})\,|\mathbf{s}\cdot\boldsymbol{\nu}|$, has a cosine distribution — this is Lambert's well known cosine law (see Fig. 8.1 (b)). The total energy emitted by the source is $\pi I_\mathrm{d}(\mathbf{r})$. Note that the factor π (instead of 2π) arises due to the projection $|\mathbf{s}\cdot\boldsymbol{\nu}|$ normal to the surface.



(a) Specular reflection

(b) Diffuse reflection

Figure 8.1: Reflection models.

Reflection can often be described approximately as a combination of specular and diffuse reflection. Thus, an accommodation coefficient α is introduced, which determines the fraction of the incident radiation that is reflected diffusely. In order to account for the possibility of photon absorption at the surface, a particle reflection coefficient R_N is used. Furthermore, a corresponding energy reflection coefficient can be introduced as $R_\mathrm{E} = R_\mathrm{N}E_\mathrm{r}/E_0$. E_r is the mean energy of the reflected particle, and E_0 is the incident energy. In the monochromatic approximation, $E_\mathrm{r} = E_0$ so $R_\mathrm{E} = R_\mathrm{N}$. This leads to the following mixed reflection boundary condition:

$$I(\mathbf{r},\mathbf{s}) = R_\mathrm{E}\left(\alpha I_\mathrm{d}(\mathbf{r},\mathbf{s}) + (1-\alpha)I_\mathrm{s}(\mathbf{r},\mathbf{s})\right), \quad \text{for} \quad \mathbf{s}\cdot\boldsymbol{\nu} < 0. \tag{8.10}$$

Table 8.1: Reflection properties of some commen fusion reactor materials. Table based on Ref. [4].

	R_{E}	α	FWHM
Beryllium	0.85	0.02	12°
Tungsten	0.80	0.10	12°
Carbon, mat	0.20	0.10	20°
Carbon, bright	0.40	0.02	5°
Steel	0.80	0.02	2°

R_{E} and α are parameters which are determined among others by the material and the surface temperature. Table 8.1, based on Ref. [4], gives some representative values for common fusion reactor materials. The radiant energy deposited on the surface then becomes

$$\begin{aligned} Q_{\mathrm{r}} &= (1-R_{\mathrm{E}})\int_{\mathbf{s}\cdot\boldsymbol{\nu}>0} I(\mathbf{r},\mathbf{s})\mathbf{s}\cdot\boldsymbol{\nu}\,\mathrm{d}\mathbf{s} \\ &= (1-R_{\mathrm{E}})\int_{\mathbf{s}\cdot\boldsymbol{\nu}>0} I(\mathbf{r},\mathbf{s})\mathbf{s}\,\mathrm{d}\mathbf{s}\cdot\boldsymbol{\nu} \\ &\equiv \mathbf{Q}_{\mathrm{r}}\cdot\boldsymbol{\nu}. \end{aligned} \tag{8.11}$$

8.2 Optimization of Radiation Transport

As in Chapter 4 and 5, the aim is to achieve as uniform a power load as possible at the divertor targets. This load now has contributions from (1) heat Q_{t} convected and conducted by the plasma, (2) potential energy due to surface recombination Q_{sr} and (3) absorbed radiation Q_{r}:

$$Q_{\mathrm{o}} = Q_{\mathrm{t}} + Q_{\mathrm{sr}} + Q_{\mathrm{r}}.$$

The cost functional is the same as in Eq. (4.1),

$$J(\Omega,\mathbf{q}) = \frac{1}{2}\int_{\mathrm{t}} (Q_{\mathrm{o}} - Q_{\mathrm{d}})^2\,\mathrm{d}\sigma. \tag{8.12}$$

The state variables $\mathbf{q}$ now also include the radiance I. The Lagrangian is further extended by adding the radiative transfer equation and its boundary conditions. Since the radiative transfer equation is in 5D phase space, also the adjoint radiance $I^*(\mathbf{r},\mathbf{s})$ and its counterpart for the boundary conditions, $I^*_{\mathrm{S}}(\mathbf{r},\mathbf{s})$, need to be phase space variables. In the Lagrangian, this means an integral over phase space:

$$\begin{aligned} L(\Omega,\mathbf{q},\mathbf{q}^*) &= L_0(\Omega,\mathbf{q},\mathbf{q}^*) + \int_\Omega \int_{4\pi} I^*(\mathbf{r},\mathbf{s})\left(S(\mathbf{r},\mathbf{s}) - \mathbf{s}\cdot\nabla I(\mathbf{r},\mathbf{s})\right)\mathrm{d}s\,\mathrm{d}\boldsymbol{\omega} \\ &+ \int_\Sigma \int_{\mathbf{s}\cdot\boldsymbol{\nu}<0} I^*_{\mathrm{S}}(\mathbf{r},\mathbf{s})\left(I(\mathbf{r},\mathbf{s}) - R_{\mathrm{E}}\left(\alpha I_{\mathrm{d}}(\mathbf{r},\mathbf{s}) + (1-\alpha)I_{\mathrm{s}}(\mathbf{r},\mathbf{s})\right)\right)\mathbf{s}\cdot\boldsymbol{\nu}\,\mathrm{d}s\,\mathrm{d}\sigma, \end{aligned} \tag{8.13}$$

with $L_0(\Omega,\mathbf{q},\mathbf{q}^*)$ the Lagrangian defined in Section 4.1, Eq. (4.3). The optimality conditions are obtained by following similar steps as in Section 3.3. Only the additional terms which arise due to the radiation transport are elaborated here.

8.2.1 Linearized Radiation Transport

Linearizing the radiative transfer equation (8.2) with respect to the state variables directly leads to

$$\mathbf{s}\cdot\nabla\delta I(\mathbf{r},\mathbf{s}) = S_{\mathbf{q}}(\mathbf{r},\mathbf{s})\delta\mathbf{q}. \tag{8.14}$$

The equation is already linear in the radiance. Linearization of the radiation source also implies that rate coefficients have to be linearized with respect to densities and temperatures. The linearized boundary conditions are

$$\delta I(\mathbf{r},\mathbf{s}) = R_{\mathrm{E}}\left(\alpha\delta I_{\mathrm{d}}(\mathbf{r},\mathbf{s}) + (1-\alpha)\delta I_{\mathrm{s}}(\mathbf{r},\mathbf{s})\right), \quad \text{for} \quad \mathbf{s}\cdot\boldsymbol{\nu} < 0, \tag{8.15}$$

with

$$\begin{aligned} \delta I_{\mathrm{d}}(\mathbf{r},\mathbf{s}) &= \frac{1}{\pi}\int_{\mathbf{s}'\cdot\boldsymbol{\nu}>0} \delta I(\mathbf{r},\mathbf{s}')\mathbf{s}'\cdot\boldsymbol{\nu}\,\mathrm{d}s', \quad \text{for} \quad \mathbf{s}\cdot\boldsymbol{\nu} < 0, &(8.16)\\ \delta I_{\mathrm{s}}(\mathbf{r},\mathbf{s}) &= \delta I(\mathbf{r},\mathbf{s} - 2(\mathbf{s}\cdot\boldsymbol{\nu})\boldsymbol{\nu}), \quad \text{for} \quad \mathbf{s}\cdot\boldsymbol{\nu} < 0. &(8.17) \end{aligned}$$

Implicitly, it is assumed that the coefficients R_{E} and α are independent of $\mathbf{q}$. The other state equations do not involve the radiance, and thus their linearization is not changed.

Linearizing the cost functional gives

$$J_{\mathbf{q}}(\Omega,\mathbf{q})\delta\mathbf{q} = \int_{\mathrm{t}} \left(Q_{\mathrm{o}} - Q_{\mathrm{d}}\right)\delta Q_{\mathrm{o}}\,\mathrm{d}\sigma. \tag{8.18}$$

The contribution from radiation in this last expression is

$$\delta Q_{\mathrm{r}} = (1 - R_{\mathrm{E}}) \int_{\mathbf{s}\cdot\boldsymbol{\nu}>0} \delta I(\mathbf{r}, \mathrm{s})\mathrm{s} \cdot \boldsymbol{\nu} \, \mathrm{ds}. \tag{8.19}$$

8.2.2 Adjoint Radiation Transport

The adjoint equations are derived by moving all differential operators in the linearized Lagrangian from the linearized state variables to the adjoint multipliers. Writing only the contributions of the linearized radiation problem explicitly, this leads to:

$$\begin{aligned}
&L_{\mathbf{q}}\delta\mathbf{q} \\
&= L_{0,\mathbf{q}}\delta\mathbf{q} + \int_\Omega \int_{4\pi} I^*(\mathbf{r}, \mathrm{s}) \left(S_{\mathbf{q}}(\mathbf{r}, \mathrm{s})\delta\mathbf{q} - \mathrm{s} \cdot \nabla \delta I(\mathbf{r}, \mathrm{s})\right) \, \mathrm{ds} \, \mathrm{d}\boldsymbol{\omega} \\
&\quad + \int_\Sigma \int_{\mathbf{s}\cdot\boldsymbol{\nu}<0} I^*_{\mathrm{S}}(\mathbf{r}, \mathrm{s}) \left(\delta I(\mathbf{r}, \mathrm{s}) - R_{\mathrm{E}} \left(\alpha \delta I_{\mathrm{d}}(\mathbf{r}, \mathrm{s}) + (1-\alpha)\delta I_{\mathrm{s}}(\mathbf{r}, \mathrm{s})\right)\right) \mathrm{s} \cdot \boldsymbol{\nu} \, \mathrm{ds} \, \mathrm{d}\sigma \\
&\quad + \int_{\mathrm{t}} (Q_{\mathrm{o}} - Q_{\mathrm{d}}) (1 - R_{\mathrm{E}}) \int_{\mathbf{s}\cdot\boldsymbol{\nu}>0} \delta I(\mathbf{r}, \mathrm{s})\mathrm{s} \cdot \boldsymbol{\nu} \, \mathrm{ds} \, \mathrm{d}\sigma \\
&= L_{0,\mathbf{q}}\delta\mathbf{q} + \int_\Omega \int_{4\pi} I^*(\mathbf{r}, \mathrm{s}) S_{\mathbf{q}}(\mathbf{r}, \mathrm{s})\delta\mathbf{q} + \delta I(\mathbf{r}, \mathrm{s})\mathrm{s} \cdot \nabla I^*(\mathbf{r}, \mathrm{s}) \, \mathrm{ds} \, \mathrm{d}\boldsymbol{\omega} \\
&\quad - \int_\Sigma \int_{4\pi} I^*(\mathbf{r}, \mathrm{s})\delta I(\mathbf{r}, \mathrm{s})\mathrm{s} \cdot \boldsymbol{\nu} \, \mathrm{ds} \, \mathrm{d}\sigma \\
&\quad + \int_\Sigma \int_{\mathbf{s}\cdot\boldsymbol{\nu}<0} I^*_{\mathrm{S}}(\mathbf{r}, \mathrm{s}) \left(\delta I(\mathbf{r}, \mathrm{s}) - R_{\mathrm{E}} \left(\alpha \delta I_{\mathrm{d}}(\mathbf{r}, \mathrm{s}) + (1-\alpha)\delta I_{\mathrm{s}}(\mathbf{r}, \mathrm{s})\right)\right) \mathrm{s} \cdot \boldsymbol{\nu} \, \mathrm{ds} \, \mathrm{d}\sigma \\
&\quad + \int_{\mathrm{t}} (Q_{\mathrm{o}} - Q_{\mathrm{d}}) (1 - R_{\mathrm{E}}) \int_{\mathbf{s}\cdot\boldsymbol{\nu}>0} \delta I(\mathbf{r}, \mathrm{s})\mathrm{s} \cdot \boldsymbol{\nu} \, \mathrm{ds} \, \mathrm{d}\sigma,
\end{aligned} \tag{8.20}$$

with $L_{0,\mathbf{q}}\delta\mathbf{q}$ from Eq. (3.43), and $\delta I_{\mathrm{d}}(\mathbf{r}, \mathrm{s})$ and $\delta I_{\mathrm{d}}(\mathbf{r}, \mathrm{s})$ from Eqs. (8.16) and (8.17). The volume integral leads to additional source terms in the adjoint plasma equations,

$$S^T_{\mathbf{q}} \int_{4\pi} I^*(\mathbf{r}, \mathrm{s}) \, \mathrm{ds}, \tag{8.21}$$

and to the adjoint radiative transfer equation

$$-\, \mathrm{s} \cdot \nabla I^*(\mathbf{r}, \mathrm{s}) = 0. \tag{8.22}$$

Since the other plasma state equations do not depend on I^*, there are no source terms in this last equation. In particular, it does not depend on the other adjoint variables, so it can be solved in a *preprocessing* step to evaluate the source terms (8.21) for the adjoint plasma equations. The boundary conditions are obtained by eliminating the boundary integrals in Eq. (8.20). Some further computations reveal that this is accomplished by the choices

$$\begin{aligned}
I^*(\mathbf{r},\mathbf{s}) &= (Q_\mathrm{o}-Q_\mathrm{d})\,(1-R_\mathrm{E})+ \\
&\quad R_\mathrm{E}\left(\alpha I^*_\mathrm{d}(\mathbf{r},\mathbf{s})+(1-\alpha)I^*_\mathrm{s}(\mathbf{r},\mathbf{s})\right), \quad \text{for} \quad \mathbf{s}\cdot\boldsymbol{\nu}>0 \quad \text{on t}, \\
I^*(\mathbf{r},\mathbf{s}) &= R_\mathrm{E}\left(\alpha I^*_\mathrm{d}(\mathbf{r},\mathbf{s})+(1-\alpha)I^*_\mathrm{s}(\mathbf{r},\mathbf{s})\right), \quad \text{for} \quad \mathbf{s}\cdot\boldsymbol{\nu}>0 \quad \text{on } \Sigma\backslash\text{t},
\end{aligned}$$

combined with

$$\begin{aligned}
I^*_\mathrm{d}(\mathbf{r},\mathbf{s}) &= -\frac{1}{\pi}\int_{\mathbf{s}'\cdot\boldsymbol{\nu}<0} \mathbf{s}'\cdot\boldsymbol{\nu} I^*(\mathbf{r},\mathbf{s}')\,\mathrm{d}\mathbf{s}', \quad \text{for} \quad \mathbf{s}\cdot\boldsymbol{\nu}>0, \\
I^*_\mathrm{s}(\mathbf{r},\mathbf{s}) &= I^*(\mathbf{r},\mathbf{s}-2(\mathbf{s}\cdot\boldsymbol{\nu})\boldsymbol{\nu}), \quad \text{for} \quad \mathbf{s}\cdot\boldsymbol{\nu}>0, \\
I^*_\mathrm{S}(\mathbf{r},\mathbf{s}) &= I^*(\mathbf{r},\mathbf{s}).
\end{aligned}$$

The derivative of the cost functional appears as a uniform adjoint radiance (i.e. independent of direction). Indeed, the cost functional only measures the total energy deposited on the target, and doesn't care where this energy is coming from. In contrast to the forward problem, the adjoint radiance is not necessarily positive, but its sign depends on the sign of $Q_\mathrm{o}-Q_\mathrm{d}$. The boundary conditions for the other adjoint plasma equations (see Appendix B) do not change, except for the terms involving $Q_\mathrm{o}-Q_\mathrm{d}$ at the targets which now also contain a contribution from radiation.

The convective term in the adjoint radiative transfer equation has switched sign compared to the forward equation, so adjoint radiation $I^*(\mathbf{r},\mathbf{s})$ moves in the $-\mathbf{s}$ direction. The same effect is visible in the boundary conditions: they have to be applied for directions $\mathbf{s}$ with $\mathbf{s}\cdot\boldsymbol{\nu}>0$. By making the substitution $\mathbf{s}\to-\mathbf{s}$ and defining $F^*(\mathbf{r},\mathbf{s})\equiv I^*(\mathbf{r},-\mathbf{s})$, equation (8.22) can be written as

$$\mathbf{s}\cdot\nabla F^*(\mathbf{r},\mathbf{s})=0, \tag{8.23}$$

with boundary conditions

$$\begin{aligned} F^*(\mathbf{r},\mathbf{s}) &= (Q_\mathrm{o} - Q_\mathrm{d})\,(1-R_\mathrm{E}) + \\ &\quad R_\mathrm{E}\left(\alpha F^*_\mathrm{d}(\mathbf{r},\mathbf{s}) + (1-\alpha)F^*_\mathrm{s}(\mathbf{r},\mathbf{s})\right), \text{ for } \mathbf{s}\cdot\boldsymbol{\nu} < 0 \quad \text{on t}, \\ F^*(\mathbf{r},\mathbf{s}) &= R_\mathrm{E}\left(\alpha F^*_\mathrm{d}(\mathbf{r},\mathbf{s}) + (1-\alpha)F^*_\mathrm{s}(\mathbf{r},\mathbf{s})\right), \text{ for } \mathbf{s}\cdot\boldsymbol{\nu} < 0 \quad \text{on } \Sigma\backslash\text{t}. \end{aligned}$$

The diffusive and reflective adjoint intensities are given by

$$\begin{aligned} F^*_\mathrm{d}(\mathbf{r},\mathbf{s}) &= \frac{1}{\pi}\int_{\mathbf{s}'\cdot\boldsymbol{\nu}>0} \mathrm{s}'\cdot\boldsymbol{\nu}F^*(\mathbf{r},\mathrm{s}')\,\mathrm{d}\mathrm{s}', \quad \text{for} \quad \mathbf{s}\cdot\boldsymbol{\nu} < 0, \\ F^*_\mathrm{s}(\mathbf{r},\mathbf{s}) &= F^*(\mathbf{r},\mathrm{s} - 2(\mathrm{s}\cdot\boldsymbol{\nu})\boldsymbol{\nu}), \quad \text{for} \quad \mathbf{s}\cdot\boldsymbol{\nu} < 0, \end{aligned}$$

while the boundary condition multiplier is

$$F^*_\mathrm{S}(\mathbf{r},\mathbf{s}) = F^*(\mathbf{r},\mathbf{s}).$$

Comparing with Equation (8.2) and boundary conditions (8.10), this means that the adjoint radiation problem can be solved for using exactly the same Monte Carlo process as the forward radiation problem, followed by the back substitution $I^*(\mathbf{r},\mathbf{s}) = F^*(\mathbf{r},-\mathbf{s})$. The source of adjoint radiation is changed from a volumetric source to a surface source on the targets, and the adjoint radiance has to be known in all cells, instead of only at the targets.

8.2.3 Design Equation

In order to compute the derivative of the cost functional (8.12), the shape sensitivity of the Lagrangian (8.13) is needed. As in Equation 4.4, the contributions to the shape derivative are split into three parts,

$$\dot{J} = \dot{J}_1 + \dot{J}_2 + \dot{J}_3, \tag{8.24}$$

where the first contribution stems from the cost functional itself, the second from the volume integrals of the state equations, and the third from the boundary condition integrals.

The expression for the shape derivative of the cost functional obtained in Eq. (4.5) can be extended quite easily. The only additional difficulty is that the integral over the directional component in Q_r, Eq. (8.11), depends on the surface normal through the limits of integration. The variation due to the change in limits of integration can be taken into account by the Leibniz integral rule. However, since the factor $\mathbf{s} \cdot \boldsymbol{\nu}$ appears in the integrand, the corresponding contributions will be zero. Thus, the shape derivative of the cost functional is

$$\begin{aligned} \dot{J}_1 &= \int_{\mathrm{t}} \underbrace{\left(\frac{1}{2} \left(Q_{\mathrm{o}} - Q_{\mathrm{d}} \right)^2 - \left(Q_{\mathrm{o}} - Q_{\mathrm{d}} \right) Q_{\mathrm{o}} \right) \kappa}_{(1)} \boldsymbol{\mathcal{V}} \cdot \boldsymbol{\nu} \, \mathrm{d}\sigma \\ &+ \int_{\mathrm{t}} \left(\underbrace{\left(Q_{\mathrm{o}} - Q_{\mathrm{d}} \right) \nabla \cdot \mathbf{Q}_{\mathrm{o}}}_{(2)} + \underbrace{\mathbf{Q}_{\mathrm{o}} \cdot \nabla_{\Sigma} \left(Q_{\mathrm{o}} - Q_{\mathrm{d}} \right)}_{(3)} \right) \boldsymbol{\mathcal{V}} \cdot \boldsymbol{\nu} \, \mathrm{d}\sigma. \quad (8.25) \end{aligned}$$

The curvature term (1) can be evaluated easily. For the contribution of radiation, it is convenient to combine terms (2) and (3) as

$$\begin{aligned} &\int_{\mathrm{t}} \left(\left(Q_{\mathrm{o}} - Q_{\mathrm{d}} \right) \nabla \cdot \mathbf{Q}_{\mathrm{r}} + \mathbf{Q}_{\mathrm{r}} \cdot \nabla_{\Sigma} \left(Q_{\mathrm{o}} - Q_{\mathrm{d}} \right) \right) \boldsymbol{\mathcal{V}} \cdot \boldsymbol{\nu} \, \mathrm{d}\sigma \\ &\qquad = \int_{\mathrm{t}} \left(\left(Q_{\mathrm{o}} - Q_{\mathrm{d}} \right) \nabla_{\boldsymbol{\nu}} \cdot \mathbf{Q}_{\mathrm{r}} + \nabla_{\Sigma} \cdot \left(\left(Q_{\mathrm{o}} - Q_{\mathrm{d}} \right) \mathbf{Q}_{\mathrm{r}} \right) \right) \boldsymbol{\mathcal{V}} \cdot \boldsymbol{\nu} \, \mathrm{d}\sigma \end{aligned}$$

Indeed, while quantities related to the plasma are most easily computed in the poloidal-radial coordinate system determined by the magnetic field, it is more natural to express radiative quantities in a normal-tangential system. The tangential divergence term can be evaluated directly by accumulating the tangential flux

$$(1 - R_{\mathrm{E}}) \int_{\mathbf{s} \cdot \boldsymbol{\nu} > 0} I(\mathbf{r}, \mathbf{s}) \mathbf{s} \cdot \boldsymbol{\tau} \, \mathrm{d}\mathbf{s} \qquad (8.26)$$

during the solution of the forward radiation problem. Here, $\boldsymbol{\tau}$ the unit vector tangential to the surface (and in the poloidal plane). The normal divergence term requires derivatives normal to the boundary.

Next, the additional volume integral in the Lagrangian (8.13) is treated. From Equation (2.18) it is seen that its contribution is zero if the radiative transfer equation is satisfied. Therefore, as was the case in Eq. (4.6), $\dot{J}_2 = 0$.

Finally, the integral in the Lagrangian due to the boundary conditions is treated. The boundary conditions have been written as flux (Neumann) conditions. However, there is a further dependence on the normal through the arguments in I_{s} and I_{d}, so expression (2.25) is applied to compute the shape derivative.

The contributions from the individual terms in

$$\int_\Sigma \int_{\mathbf{s}\cdot\boldsymbol{\nu}<0} I_\mathrm{S}^*(\mathbf{r},\mathbf{s}) \left(\underbrace{I(\mathbf{r},\mathbf{s})}_{(a)} - R_\mathrm{E} \left(\underbrace{\alpha I_\mathrm{d}(\mathbf{r},\mathbf{s})}_{(b)} + \underbrace{(1-\alpha) I_\mathrm{s}(\mathbf{r},\mathbf{s})}_{(c)} \right)\right) \mathbf{s}\cdot\boldsymbol{\nu}\,\mathrm{d}\mathbf{s}\,\mathrm{d}\sigma \tag{8.27}$$

are elaborated one by one.

For term (a), elaboration gives

$$\dot{J}_{3,a} = \int_\Sigma \nabla \cdot \left(\int_{\mathbf{s}\cdot\boldsymbol{\nu}<0} I(\mathbf{r},\mathbf{s}) I^*(\mathbf{r},\mathbf{s}) \mathbf{s}\,\mathrm{d}\mathbf{s} \right) \boldsymbol{\mathcal{V}}\cdot\boldsymbol{\nu}\,\mathrm{d}\sigma. \tag{8.28}$$

The dependence of the limits of integration on $\boldsymbol{\nu}$ prevents further elaboration of this term. Unfortunately, this term requires complete knowledge of the forward radiation on the boundary. This issue is intrinsic to the problem. Since the forward radiation problem is 5D (4D in case of toroidal symmetry), it can be expected that also the sensitivities will have the same dimensionality.

Term (b) in Eq. (8.27) stems from diffuse reflection. Applying Equation (2.25) leads to

$$\begin{aligned} \dot{J}_{3,b} &= \int_\Sigma \int_{\mathbf{s}\cdot\boldsymbol{\nu}<0} R_\mathrm{E}\alpha\, I^*(\mathbf{r},\mathbf{s}) I_\mathrm{d}(\mathbf{r}) \mathbf{s}\cdot\boldsymbol{\nu}\,\mathrm{d}\mathbf{s}\,\kappa \boldsymbol{\mathcal{V}}\cdot\boldsymbol{\nu}\,\mathrm{d}\sigma \\ &- \int_\Sigma R_\mathrm{E}\alpha \left(I_\mathrm{d}(\mathbf{r}) \nabla_{\boldsymbol{\nu}} \cdot \int_{\mathbf{s}\cdot\boldsymbol{\nu}<0} I^*(\mathbf{r},\mathbf{s})\mathbf{s}\,\mathrm{d}\mathbf{s} - I_\mathrm{d}^*(\mathbf{r}) \nabla_{\boldsymbol{\nu}} \cdot \int_{\mathbf{s}\cdot\boldsymbol{\nu}>0} I(\mathbf{r},\mathbf{s})\mathbf{s}\,\mathrm{d}\mathbf{s} \right) \boldsymbol{\mathcal{V}}\cdot\boldsymbol{\nu}\,\mathrm{d}\sigma \\ &- \int_\Sigma R_\mathrm{E}\alpha\, \nabla_\Sigma \cdot \left(I_\mathrm{d}(\mathbf{r}) \int_{\mathbf{s}\cdot\boldsymbol{\nu}<0} I^*(\mathbf{r},\mathbf{s})\mathbf{s}\,\mathrm{d}\mathbf{s} - I_\mathrm{d}^*(\mathbf{r}) \int_{\mathbf{s}\cdot\boldsymbol{\nu}>0} I(\mathbf{r},\mathbf{s})\mathbf{s}\,\mathrm{d}\mathbf{s} \right) \boldsymbol{\mathcal{V}}\cdot\boldsymbol{\nu}\,\mathrm{d}\sigma. \end{aligned} \tag{8.29}$$

It is remarked that all these terms can be evaluated without the need of the complete radiation distribution. This is an interesting consequence of the isotropic nature of diffuse reflection.

Lastly, the term due to specular reflection in investigated. Elaborating with expression (2.25) now gives

$$\begin{aligned} \dot{J}_{3,c} = &- \int_\Sigma 4R_\mathrm{E}(1-\alpha) \int_{\mathbf{s}\cdot\boldsymbol{\nu}<0} I^*(\mathbf{r},\mathbf{s}) \boldsymbol{\nu}\cdot\nabla_\mathbf{s} I_\mathrm{s}(\mathbf{r},\mathbf{s}) \mathbf{s}\cdot\boldsymbol{\nu}\,\mathrm{d}\mathbf{s}\,\kappa\boldsymbol{\mathcal{V}}\cdot\boldsymbol{\nu}\,\mathrm{d}\sigma \\ &- \int_\Sigma R_\mathrm{E}(1-\alpha)\nabla\cdot \int_{\mathbf{s}\cdot\boldsymbol{\nu}<0} I^*(\mathbf{r},\mathbf{s}) I_\mathrm{s}(\mathbf{r},\mathbf{s})\mathbf{s}\,\mathrm{d}\mathbf{s}\,\boldsymbol{\mathcal{V}}\cdot\boldsymbol{\nu}\,\mathrm{d}\sigma \\ &+ \int_\Sigma 2R_\mathrm{E}(1-\alpha)\nabla_\Sigma \cdot \int_{\mathbf{s}\cdot\boldsymbol{\nu}<0} \nabla_\mathbf{s} I_\mathrm{s}(\mathbf{r},\mathbf{s})\cdot\boldsymbol{\nu}\mathbf{s} + \mathbf{s}\cdot\boldsymbol{\nu}\nabla_\mathbf{s} I_\mathrm{s}(\mathbf{r},\mathbf{s})\,\mathrm{d}\mathbf{s}\,\boldsymbol{\mathcal{V}}\cdot\boldsymbol{\nu}\,\mathrm{d}\sigma \end{aligned}$$

As in term (a), it is not possible to eliminate completely the directional dependence for any of these integrals. Even the 5D gradient of the radiance is required in order to evaluate the sensitivity. This could have been expected based on the nature of specular reflection.

8.3 Implementation Aspects

Implementing the radiative transfer problem with a Monte Carlo code follows the general sampling procedure described in Section 6.1.2. However, since no scattering and no absorption of the photons is taken into account (see Eqs. (8.2) and (8.22)), the problem is relatively simple. Furthermore, the problem is assumed monochromatic, so it is convenient to work with (adjoint) ‘energy parcels’ that simply ‘bounce around’ between the walls of the vessel, while their energy is gradually deposited on these walls. The main elements of the code involve

1. Source sampling
2. Photon or ‘energy parcel’ tracing
3. Boundary conditions

Since the forward and adjoint radiative transfer equations are again very similar, these three elements are discussed for both codes together.

Source sampling

In order to sample the source of radiation, a distribution function for the source has to be constructed. Then, sampling can be done using Eq. (6.17).

In the forward problem, radiation is emitted isotropically from the cells. First, a random number ρ_0 determines from which cell the energy parcel is launched by sampling the discrete density function $f_\mathrm{c}(i) = |S(i)| \,/\, \sum_j |S(j)|$. $S(i)$ is the total amount of energy radiated from cell i. All particles are launched from the cell centers. Next, a random direction from the unit sphere is drawn. Therefore, the density function

$$f(\theta, \phi) = \frac{1}{4\pi} \sin\theta \,\mathrm{d}\theta \,\mathrm{d}\phi, \qquad 0 \leq \theta \leq \pi, \qquad 0 \leq \phi \leq 2\pi$$

has to be sampled [115]. This can be done by two additional random numbers such that

$$\begin{aligned}\rho_1 &= \int_0^\theta \int_0^{2\pi} f(t,\phi)\,\mathrm{d}\phi\,\mathrm{d}t \;=\; \frac{1}{2}\int_0^\theta \sin(t)\,\mathrm{d}t,\\ \rho_2 &= \int_0^\phi \int_0^{\pi} f(\theta,t)\,\mathrm{d}\theta\,\mathrm{d}t \;=\; \int_0^\phi \frac{1}{2\pi}\,\mathrm{d}t,\end{aligned}$$

or $\theta = \cos^{-1}(1-2\rho_1)$ and $\phi = 2\pi\rho_2$.

In the adjoint problem, the radiation is emitted with uniform intensity from the targets. Again, first a random number ρ_0 determines from which cell face the particle is to be launched by sampling the discrete density function $|S^*(i)| / \sum_j |S^*(j)|$. The summation is now only along the target faces. Then, a direction has to be sampled that corresponds to a uniform intensity I^*. In terms of adjoint energy flux emitted in a solid angle around $\mathbf{s}$, this means $\mathrm{d}E^* = I^*\cos(\theta)\mathrm{d}\mathbf{s} = I^*\cos(\theta)\sin(\theta)\mathrm{d}\theta\mathrm{d}\phi$, while the total energy emitted from the surface is

$$E_\mathrm{t}^* = \int_0^{2\pi}\int_0^{\frac{\pi}{2}} I^*\cos(\theta)\sin(\theta)\,\mathrm{d}\theta\,\mathrm{d}\phi = \pi I^*.$$

The density function to be sampled is thus

$$f^*(\theta,\phi) = \frac{\mathrm{d}E^*}{E_\mathrm{t}^*} = \frac{1}{\pi}\cos(\theta)\sin(\theta)\,\mathrm{d}\theta\,\mathrm{d}\phi, \qquad 0 \le \theta \le \frac{\pi}{2}, \qquad 0 \le \phi \le 2\pi.$$

The angles θ and ϕ can be sampled by

$$\rho_1 = \int_0^\theta \int_0^{2\pi} f^*(t,\phi)\,\mathrm{d}\phi\,\mathrm{d}t \;=\; \int_0^\theta \sin(2t)\,\mathrm{d}t, \tag{8.30}$$

$$\rho_2 = \int_0^\phi \int_0^{\frac{\pi}{2}} f^*(\theta,t)\,\mathrm{d}\theta\,\mathrm{d}t \;=\; \int_0^\phi \frac{1}{2\pi}\,\mathrm{d}t, \tag{8.31}$$

so $\theta = \sin^{-1}\left(\sqrt{\rho_1}\right)$ and $\phi = 2\pi\rho_2$. Note that the cosine law does not imply that most particles leave in the direction normal to the surface [49]. Due to the 3D nature of the problem, the density function for the polar angle is $\sin(2\theta)$, so most particles leave at an angle of 45° compared to the normal.

Photon tracing

In the radiation problem, the particles move between the true vessel walls. The vessel is assumed to be described by a piecewise linear representation in the poloidal plane. In the toroidal direction, the walls are perfectly circular. Parametrizing the particle trajectory by

$$\mathbf{X}_{\mathrm{p}}(t) = \mathbf{X}_0 + t\mathbf{s}, \qquad \mathbf{X}_0 = \begin{pmatrix} R_0 \\ 0 \\ Z_0 \end{pmatrix}, \qquad t > 0,$$

with $\mathbf{X}_0$ the initial position of the photon, and each segment of the vessel by

$$\mathbf{X}_{\mathrm{v}}(u, \phi) = \begin{pmatrix} ((1-u)R_1 + uR_2)\cos\phi \\ ((1-u)R_1 + uR_2)\sin\phi \\ (1-u)Z_1 + uZ_2 \end{pmatrix}, \qquad 0 \leq u \leq 1, \qquad 0 \leq \phi \leq 2\pi,$$

where $\mathbf{X}_1 = (R_1, 0, Z_1)^T$ and $\mathbf{X}_2 = (R_2, 0, Z_2)$ are the vertices of the vessel segment in the poloidal plane, see Fig. 8.2, the intersection between the trajectory of the photon and the vessel segment can be found by solving the system

$$\mathbf{X}_{\mathrm{p}}(t) = \mathbf{X}_{\mathrm{v}}(u, \phi)$$

for t, u and ϕ. This equation is solved for all vessel segments. Then, the flight is carried out only until the nearest intersection, i.e. the one with the smallest value of t. In this way, the full 3D toroidal geometry is correctly taken into account. Since toroidal symmetry is assumed, each particle is rotated back to the poloidal plane with an angle $-\phi$ after its flight.

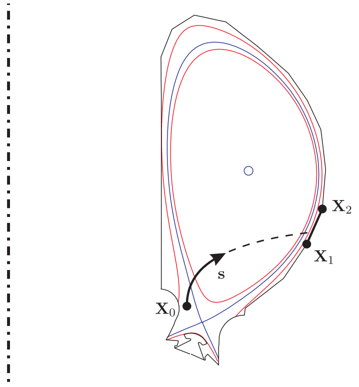


Figure 8.2: Computation of intersection between photon trajectory and a vessel segment. In the poloidal plane, the photon trajectory does not necessarily appear as straight.

For the forward radiation problem, only the intersections with the true vessel are required in the simulation. At these intersections, the fluxes to the wall can be computed. This will be further discussed in the paragraph on boundary conditions below. For the adjoint problem, on the other hand, also the radiance integrated over each cell of the plasma domain is required in order to compute the source terms (8.21). Therefore, also all intersections with cell faces crossed during the flight of the photon have to be computed. Track-length estimators are then used to accumulate the radiance cell by cell. It is remarked that in practice this makes the adjoint simulation computationally much more demanding because the number of cell faces is much larger than the number of vessel segments[1]. Finally, in order to evaluate design sensitivities, derivatives of the (forward and adjoint) fluxes at the target are required. Tangential derivatives can be computed based on the fluxes themselves. For the normal derivatives, the fluxes are also computed at additional surfaces translated along the inward normal of the vessel segments.

Boundary conditions

At each wall, a mixed reflection condition of the form (8.10) is applied. First of all, the energy of the parcel is reduced by the energy reflection coefficient R_E (physically, this corresponds to a fraction of the incident photons being absorbed). The fraction $1 - R_\mathrm{E}$ of the energy is absorbed locally by the wall. Also the local energy flux vector can be computed directly. Then, a random number ρ_0 determines which type of reflection will occur. In case of specular reflection ($\rho_0 > \alpha$), Eq. (8.6) is used to compute the direction of the reflected radiation. In case of diffuse reflection, the angle of reflection is sampled from a cosine distribution as in Eqs. (8.30)–(8.31).

It is remarked that at least one surface must have an energy reflection coefficient smaller than 1 in order to have a well posed problem. Furthermore, if there are no perfectly absorbing surfaces, the parcels will travel infinitely long. However, their energy will be reduced due to successive reflections. Therefore, a minimum weight w_min can be specified by the user, so that any parcel with $w < w_\mathrm{min}$ is removed from the simulation. Note that this minimum weight also determines the accuracy on the power balance.

8.4 Numerical Validation

In this Section, the implementation of the expressions for the shape sensitivities derived in Section 8.2 is tested. To this end, test problems are considered which

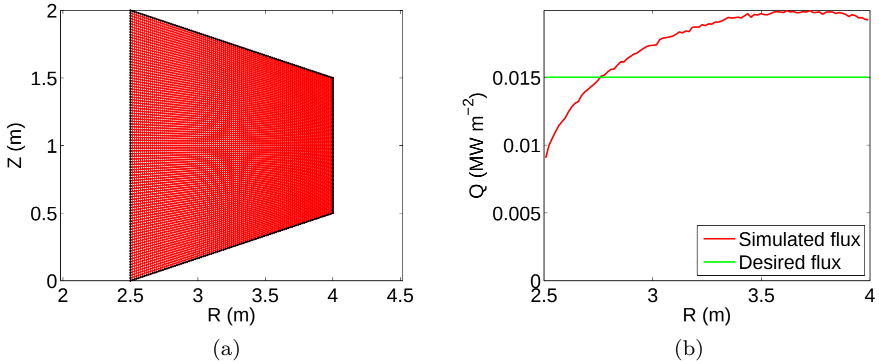
---

[1] In case absorption and scattering are taken into account, also the forward problem requires the intersections with all cell faces and becomes equally demanding as the adjoint problem.

include radiation transport only. By eliminating the coupling with the plasma, these problems pose the hardest numerical test for the Monte Carlo routines. Furthermore, they will allow to identify any issues related to the statistical noise from the Monte Carlo procedure. Problems including both edge plasma and radiation models are considered in Section 8.5.

Shape sensitivities of radiation load

In a first test problem, the shape sensitivities computed with the continuous adjoint approach and shape sensitivity analysis are compared to the ones obtained with finite differences on the domain shape. The simple toroidal geometry illustrated in Fig. 8.3 (a) is considered. The sizes are representative of a small sized tokamak. Inside the domain, plasma density and temperature are assumed constant: $n = 1 \cdot 10^{19}$ m^{-3}, $c_z = 0.5\%$ and $T = 25$ eV. Furthermore, for Deuterium $E_{\mathrm{i,r}} + E_{\mathrm{r,r}} = 25$ eV. As a result, also the source of radiation is constant in the domain. At the targets (top and bottom boundaries), reflection properties of mat Carbon are used, see Table 8.1. However, since the shape sensitivity for specular reflection is hard to compute, perfect diffuse reflection is assumed for simplicity ($\alpha = 1$). The outer wall (right boundary) is Beryllium. Here, specular reflection is not a problem because it is a fixed boundary. Finally, the inner wall is perfectly absorbing. This is done to provoke a significant variation of the radiation load along the targets. Indeed, since the radiation in a uniformly emitting cavity with reflecting walls is very uniform, incorporating absorbing inner walls will artificially increase the challenge of the test case.



(a) (b)

Figure 8.3: (a) Geometry of the test radiation test problem. (b) Simulated and desired fluxes for which the shape sensitivities are assessed.

Assuming the radiation will be fairly uniform, the expressions in the shape sensitivities can be simplified further. In particular, Eq. (8.28) can be reduced to

$$\begin{aligned}\dot{J}_{3,a} &= \int_\Sigma \nabla \cdot \left(\int_{\mathbf{s}\cdot\boldsymbol{\nu}<0} I(\mathbf{r},\mathbf{s}) I^*(\mathbf{r},\mathbf{s}) \mathbf{s}\, \mathrm{d}\mathbf{s} \right) \boldsymbol{\mathcal{V}} \cdot \boldsymbol{\nu}\, \mathrm{d}\sigma \\ &\approx \int_\Sigma \int_{\mathbf{s}\cdot\boldsymbol{\nu}<0} \nabla \cdot \left(I(\mathbf{r},\mathbf{s}) I^*(\mathbf{r},\mathbf{s}) \mathbf{s} \right)\, \mathrm{d}\mathbf{s} \boldsymbol{\mathcal{V}} \cdot \boldsymbol{\nu}\, \mathrm{d}\sigma \\ &= \int_\Sigma \int_{\mathbf{s}\cdot\boldsymbol{\nu}<0} I^*(\mathbf{r},\mathbf{s}) \mathbf{s} \cdot \nabla \left(I(\mathbf{r},\mathbf{s}) \right)\, \mathrm{d}\mathbf{s} \boldsymbol{\mathcal{V}} \cdot \boldsymbol{\nu}\, \mathrm{d}\sigma \\ &= \int_\Sigma S(\mathbf{r}) \int_{\mathbf{s}\cdot\boldsymbol{\nu}<0} I^*(\mathbf{r},\mathbf{s})\, \mathrm{d}\mathbf{s} \boldsymbol{\mathcal{V}} \cdot \boldsymbol{\nu}\, \mathrm{d}\sigma,\end{aligned}$$

while (8.29) simplifies to

$$\begin{aligned}\dot{J}_{3,b} = &\int_\Sigma \int_{\mathbf{s}\cdot\boldsymbol{\nu}<0} R_\mathrm{E} \alpha\, I^*(\mathbf{r},\mathbf{s}) I_\mathrm{d}(\mathbf{r}) \mathbf{s} \cdot \boldsymbol{\nu}\, \mathrm{d}\mathbf{s}\, \kappa \boldsymbol{\mathcal{V}} \cdot \boldsymbol{\nu}\, \mathrm{d}\sigma \\ -&\int_\Sigma R_\mathrm{E} \alpha \left(I_\mathrm{d}(\mathbf{r}) \nabla_{\boldsymbol{\nu}} \cdot \int_{\mathbf{s}\cdot\boldsymbol{\nu}<0} I^*(\mathbf{r},\mathbf{s}) \mathbf{s}\, \mathrm{d}\mathbf{s} - I^*_\mathrm{d}(\mathbf{r}) \nabla_{\boldsymbol{\nu}} \cdot \int_{\mathbf{s}\cdot\boldsymbol{\nu}>0} I(\mathbf{r},\mathbf{s}) \mathbf{s}\, \mathrm{d}\mathbf{s} \right) \boldsymbol{\mathcal{V}} \cdot \boldsymbol{\nu}\, \mathrm{d}\sigma \\ -&\int_\Sigma R_\mathrm{E} \alpha\, \nabla_\Sigma \cdot \left(I_\mathrm{d}(\mathbf{r}) \int_{\mathbf{s}\cdot\boldsymbol{\nu}<0} I^*(\mathbf{r},\mathbf{s}) \mathbf{s}\, \mathrm{d}\mathbf{s} - I^*_\mathrm{d}(\mathbf{r}) \int_{\mathbf{s}\cdot\boldsymbol{\nu}>0} I(\mathbf{r},\mathbf{s}) \mathbf{s}\, \mathrm{d}\mathbf{s} \right) \boldsymbol{\mathcal{V}} \cdot \boldsymbol{\nu}\, \mathrm{d}\sigma \\ \approx &\int_\Sigma R_\mathrm{E} \alpha \pi I^*_\mathrm{d}(\mathbf{r}) \left(2S(\mathbf{r}) - I_\mathrm{d}(\mathbf{r}) \kappa \right) \boldsymbol{\mathcal{V}} \cdot \boldsymbol{\nu}\, \mathrm{d}\sigma \\ -&\int_\Sigma R_\mathrm{E} \alpha \left(\int_{\mathbf{s}\cdot\boldsymbol{\nu}<0} I^*(\mathbf{r},\mathbf{s}) \mathbf{s}\, \mathrm{d}\mathbf{s} \cdot \nabla_\Sigma I_\mathrm{d}(\mathbf{r}) - \int_{\mathbf{s}\cdot\boldsymbol{\nu}>0} I(\mathbf{r},\mathbf{s}) \mathbf{s}\, \mathrm{d}\mathbf{s} \cdot \nabla_\Sigma I^*_\mathrm{d}(\mathbf{r}) \right) \boldsymbol{\mathcal{V}} \cdot \boldsymbol{\nu}\, \mathrm{d}\sigma.\end{aligned}$$

Analogously, the terms involving the normal and tangential divergence of $\mathbf{Q}_\mathrm{r}$ in Eq. (8.25) can be combined and approximated as

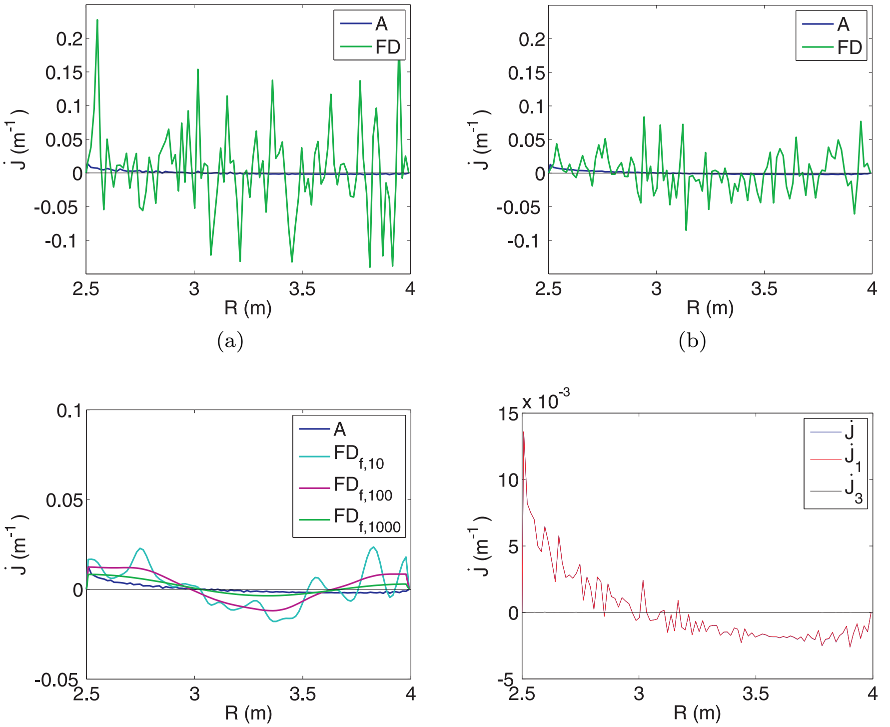
$$\begin{aligned}\nabla \cdot \mathbf{Q}_\mathrm{r} &= \nabla \cdot (1 - R_\mathrm{E}) \int_{\mathbf{s}\cdot\boldsymbol{\nu}>0} I(\mathbf{r},\mathbf{s}) \mathbf{s}\, \mathrm{d}\mathbf{s} \\ &\approx (1 - R_\mathrm{E}) \int_{\mathbf{s}\cdot\boldsymbol{\nu}>0} \mathbf{s} \cdot \nabla I(\mathbf{r},\mathbf{s})\, \mathrm{d}\mathbf{s} \\ &= 2\pi (1 - R_\mathrm{E}) S(\mathbf{r}).\end{aligned}$$

While these expressions can be evaluated easily based on boundary data only, their main advantage is that they require either no derivatives or only tangential derivatives of Monte Carlo fluxes, so they are much less prone to statistical fluctuations. These expressions have been implemented instead of the complete sensitivities given in Section 8.2.3.

The shape sensitivities of the test problem are shown in Fig. 8.4. In figure (a), 10^7 histories were used in forward and adjoint Monte Carlo codes. For the finite difference method, correlation sampling was used. In contrast to the results presented in Chapter 7, the forward approach with correlation sampling is now much more sensitive to statistical fluctuations than the continuous adjoint approach. In the 1D problem, a small perturbation of the domain length leads to a corresponding small perturbation of every single neutral trajectory and every single neutral trajectory in turn directly influences the sources in (almost) all cells. Therefore, the statistical properties of the shape sensitivities obtained with finite differences were very good. This is no longer true in 2D. Since not every photon strikes every cell face along the target, the perturbation of a single design variable is not as strongly coupled to the change in radiative energy deposited along the entire target. Since the change in energy along the entire target enters the cost functional, this change will now depend quite arbitrarily on the specific photons whose trajectory was influenced by the design perturbation. Judging from the statistical fluctuations on the forward shape sensitivity, approximately 100 times more particles are needed now to achieve the same accuracy as the continuous adjoint method. In Fig. 8.4 (b), the finite difference approach is repeated with 10^8 histories. The statistical noise is significantly reduced, but still very large compared to the continuous approach. Therefore, it is concluded that for this problem the continuous adjoint method is more robust with respect to statistical fluctuations than the discrete one. On the down side, this also means that it is very hard to validate the sensitivities computed with the continuous adjoint method. In Fig. 8.4 (c), the finite difference gradient computed with 10^8 histories is smoothed with a top hat filter. While there is qualitative agreement between the adjoint and filtered gradients, it is difficult to be confident about the result. In particular, the accuracy of less important terms in the shape sensitivity is very hard to assess. Fig. 8.4 (d) shows that in fact the adjoint sensitivity is completely determined by the direct derivative of the cost functional, $\dot{J}_1$ in Eq. (8.24). Therefore, the adjoint solution is not very important in this case.

One-shot optimization for radiation problems

In order to validate the optimization procedure itself, the test geometry with straight targets described in the previous paragraph is considered as the optimal solution to the optimization problem. The radiation load at the targets computed with 10^7 histories is used as the desired flux Q_{d}. Next, a sine perturbation is



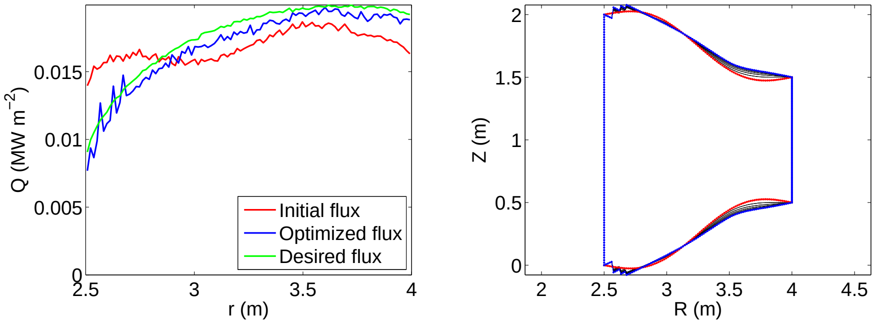
(a) (b)

(c) (d)

Figure 8.4: Comparison of continuous adjoint and finite difference shape sensitivities, (a) using 10^7 histories and (b) using 10^8 histories. Figure (c) shows the finite difference gradient of figure (b), but filtered 10, 100, and 1000 times with a top hat filter. In figure (d), the individual components of the adjoint shape sensitivity are shown (10^7 histories).

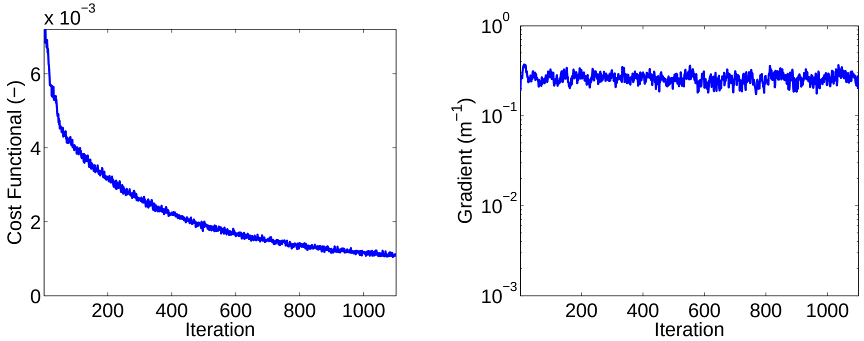
applied to the target shape, and it is tested whether the optimization algorithm finds its way back to the straight targets. During the optimization, the number of histories used in the simulation is only 10^6, i.e. a factor of 10 lower than the number of histories used to compute the reference solution. In this way, the algorithm can be tested at relatively large statistical noise. Referring to Chapter 7, uncorrelated sampling is used. In every iteration of the one-shot procedure, new random numbers are used in the forward and adjoint Monte Carlo codes.

Several tests have been performed. First, the adjoint shape sensitivity was directly used in the design update, without smoothing or filtering. Although the method seems to converge for the radiation problem, the resulting geometry with high frequency oscillations of the boundary is unacceptable if the edge plasma equations are to be solved on this domain.



(a) Initial and optimized target fluxes

(b) Sequence of domains



(c) Cost functional

(d) L1-norm of the shape sensitivity

Figure 8.5: One-shot optimization with gradient smoothing.

In a next step, Sobolev smoothing as introduced in Eq. (2.37) is used. The results of this one-shot run are presented in Figure 8.5. The design relaxation factor used in the updates was 0.1 m^2. Note that this relaxation factor is much larger than the allowed relaxation factors if the edge plasma model has to be solved on the deforming domain. The typical signature of uncorrelated sampling is noticed: despite the presence of statistical fluctuations, the cost functional is gradually reduced. Due to the large statistical fluctuations, the decrease in shape sensitivity is not even visible. The smoothing removes much of the spatial oscillations, but some artefacts remain. In this case, the optimizer does not converge to the initial configuration with straight targets but is trapped in a local optimum. Some sawtooth structures on the target have appeared. While the slope of the individual ‘teeth’ is very close to the slope of the initial targets, the optimizer did not manage to ‘merge’ these teeth into one straight target.

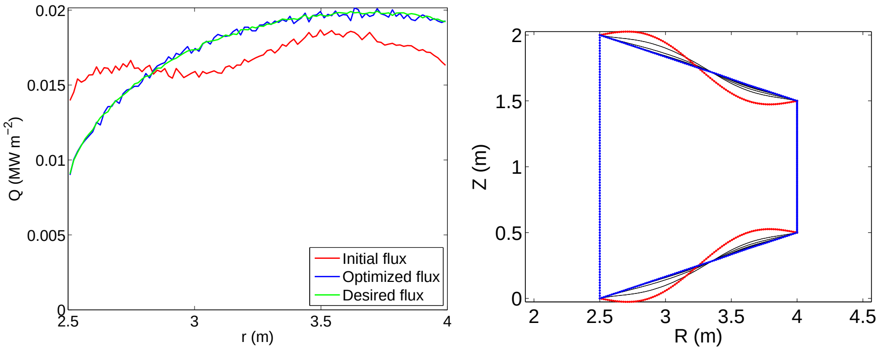
In order to further improve the performance, the optimization procedure is repeated with gradient filtering and smoothing. First, a simple top hat filter is used to average out the statistical fluctuations. In the discrete implementation, the filtered gradient $\hat{\mathcal{G}}$ is computed as

$$\hat{\mathcal{G}}_i = \frac{1}{4}\mathcal{G}_{i-1} + \frac{1}{2}\mathcal{G}_i + \frac{1}{4}\mathcal{G}_{i+1}, \tag{8.32}$$

where the face index i runs along the targets. Depending on the size of the statistical fluctuations, this filter is applied repeatedly. The Sobolev smoothing introduced in Eq. (2.37) is then used in a subsequent step on the filtered gradient $\hat{\mathcal{G}}$.

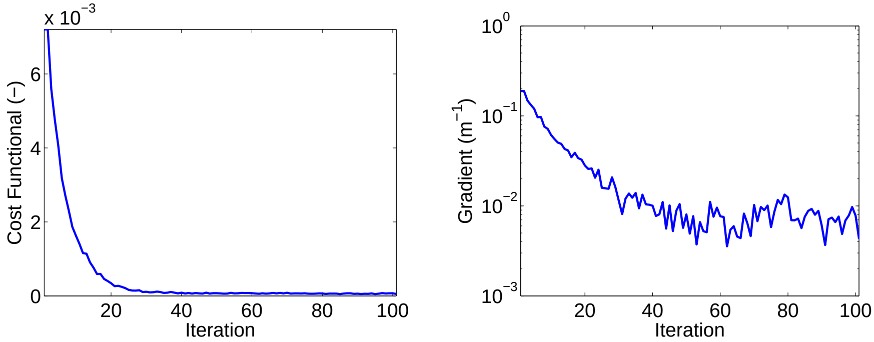
Figure 8.6 shows the results of one-shot optimization with gradient filtering and smoothing. In every iteration, the filter of Eq. (8.32) is applied 100 times. Since the filtering removes high frequency oscillations, an even larger relaxation factor of 10 m^2 is allowed. In Figure (a), the initial and optimized fluxes are compared to the desired flux. Figure (b) shows a sequence of domain shapes from the one-shot simulation. The optimizer quickly converges to the initial straight target configuration. This time, the domains are very smooth throughout the procedure, but still oscillate around the final solution.

It is remarked that although reduced formulas for the shape sensitivities are used, the optimization algorithm is still very effective. This can be regarded as an advantage of the continuous adjoint approach through shape sensitivity analysis. Clean analytical formulas are obtained, which can be manipulated based on physical insight. For the problems under consideration, this has resulted in reliable sensitivities with good statistical properties.



(a) Initial and optimized target fluxes

(b) Sequence of domains



(c) Cost functional

(d) L1-norm of the shape sensitivity

Figure 8.6: One-shot optimization with gradient filtering and smoothing.

8.5 Optimization of Total Target Power Load

In this section, the optimization of the total energy load is studied in realistic divertor geometry. The test problem defined in Chapter 5, Section 5.3, is picked up again. In order to simulate the radiation load, the vessel is added to the geometry, see Fig. 8.7. For simplicity, is assumed that the dome corresponds to the last simulated flux surface in the private flux area. The reflection properties of the different parts of the vessel are taken from Table 8.1. However, always FWHM is set to zero. The main chamber wall (up to the position where the ‘simulated’ targets start, i.e. including a small part of the baffles) is Beryllium, the dome Tungsten, and the targets mat Carbon. As in Section 8.4, perfect diffuse reflection is assumed at the targets.

In the test case, the radiation stems from Carbon and Deuterium. Both radiate mainly in the VUV-range (vacuum ultraviolet), at wavelengths of 120–130 nm. The radiation can thus be approximated by the monochromatic model. Radiation from Carbon is predominantly from the C^{4+}-state. For Deuterium, the radiation comes mainly from the Lyman-α line at approximately 120 nm.

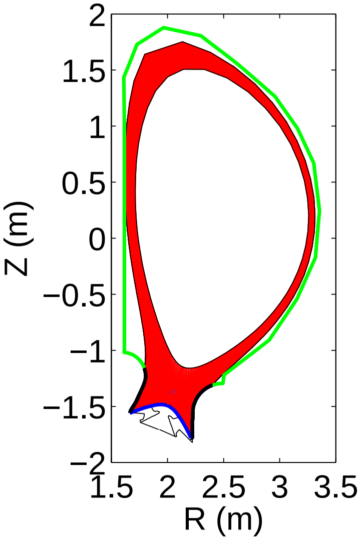
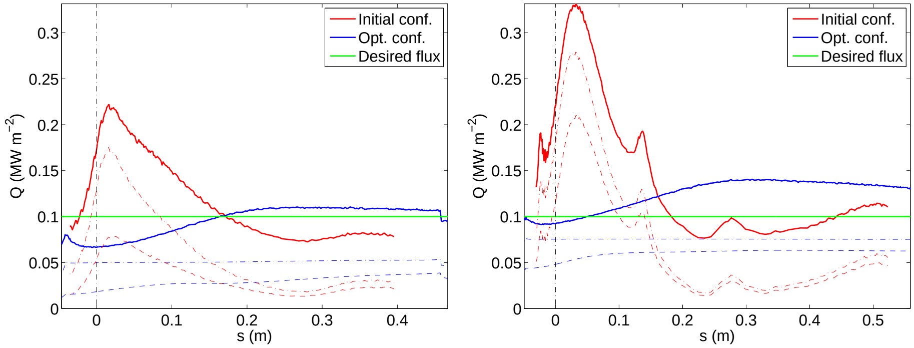


Figure 8.7: Geometry used for the simulation of the plasma and radiation. Details of the dome are neglected. The plasma grid is colored in red. The vessel segments are colored according to the material used. Black: Carbon, blue: Tungsten, green: Beryllium.

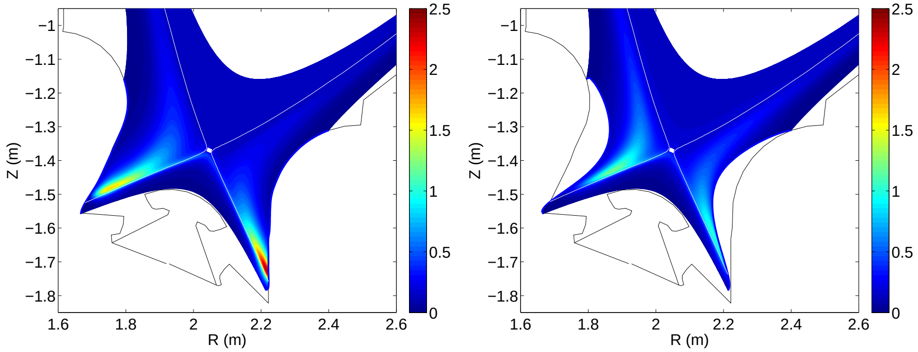
Previously, this divertor had been optimized for uniform loading without taking into account the radiative energy. Figure 8.8 shows the total energy fluxes to the inner and outer targets for initial configuration and the one optimized in Chapter 5, now including also the radiation contribution to the deposited energy. The radiation fluxes have been obtained using 10^7 histories. In this test problem, the radiative energy incident on the targets is not peaked at the separatrix strike points. Although the radiation source density is highest in the divertor area, Fig. 8.9, due to the large volume over 75% of the radiant energy still has its origin outside of the divertor. Combined with the targets tilted away from the core in both initial and optimized configurations, mainly the upper parts of the target and baffles are exposed to radiation. While the optimization of the load in Chapter 5 succeeded in increasing the load to the baffles and reducing the load at the strike point, the critical area now appears to have moved away from the strike points to the baffles. Thus, in this particular test case a trade-off can be made between heat convection and conduction, surface recombination and radiation. The aim in this section is to bring the total load as close to 0.1 MW m^{-2} as possible. It is remarked that in larger machines such as ITER, a more significant fraction of the radiation may originate in the divertor volume. Then, the radiative energy will lead to an increased peaking of the load around the strike points, see for example Ref. [51].



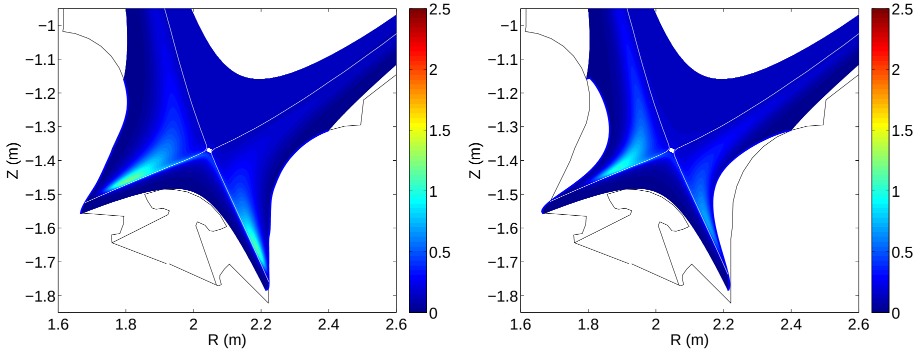
(a) Inner target (b) Outer target

Figure 8.8: Total energy fluxes to targets for scaled ITER F57 divertor (label: 'initial configuration'), and design optimized in Chapter 5 (label: 'optimized configuration'). The dashed lines represent convected and conducted energy, the dash-dotted lines include the contribution of surface recombination and the solid lines additionally include the radiative load.

When combining the edge plasma and radiation models in a one-shot optimization loop, it is important to remark that despite the fact that radiation is computed in a postprocessing step, it requires over 20% of the computational time of a complete edge plasma simulation in this case. Therefore, it is unfeasible to perform high accuracy (adjoint) radiation simulations in every iteration of the one-shot loop. On the other hand, the results of Section 8.4 indicate that much larger design steps are allowed for the radiation problem than for the edge plasma model, so the radiation evolves on a slower 'design time' scale. Therefore, the (adjoint) radiation problem is solved only every 2000 iterations of the one-shot loop, where the time steps and relaxation factors are the same as in Chapter 5. In intermediate iterations, it is assumed that the radiation load stays constant. This signature is clearly visible in the behavior of the cost functional and shape sensitivity during optimization, see Figs. 8.10 (a) and (b). The residuals of state and adjoint equations are also shown in the figure. Initially, the shape sensitivity was smoothed by applying the top hat filter of



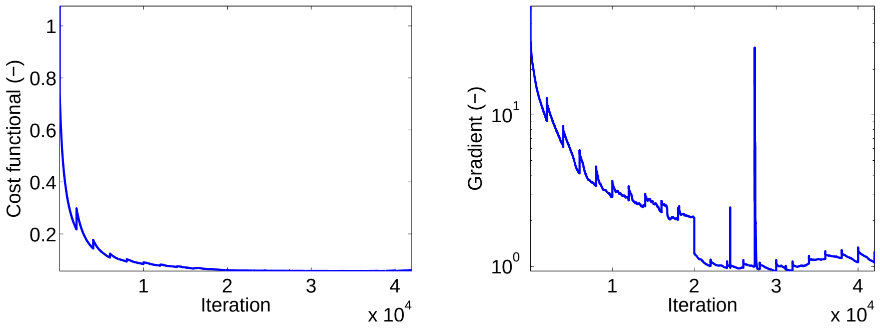
(a) Total radiation source (MW m^{-3}) (b) Total radiation source (MW m^{-3})



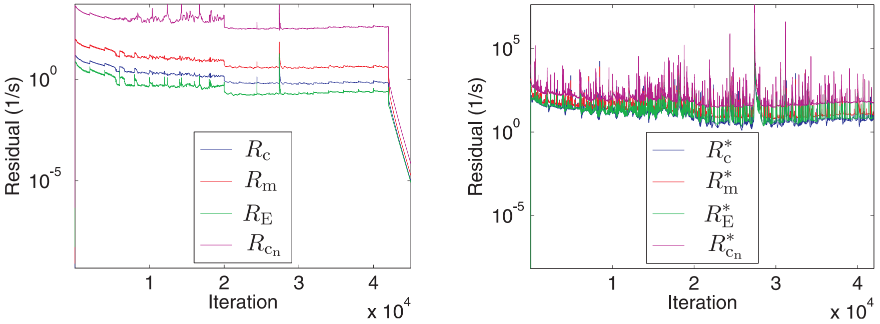
(c) Radiation source impurities (MW m^{-3}) (d) Radiation source impurities (MW m^{-3})

Figure 8.9: Radiation sources in initial configuration (left figures) and in configuration optimized in Chapter 5 (right figures).

Eq. (8.32) 10 times. At the sudden drop in residuals after 20.000 iterations, this number is increased to 100. As in the previous section, the contribution of the radiation boundary conditions in $\dot{J}_3$ to the total shape sensitivity is two or three orders of magnitude smaller than the other terms, and therefore is practically of no significance. This is not the same as saying the adjoint radiation problem should not be solved. Indeed, the adjoint radiance I^* contributes to the source terms in the adjoint plasma equations, see Eq. (8.21). These terms are of the same order of magnitude as the other sources in the adjoint plasma equations and cannot be neglected. Physically, these terms are related to the change in total radiated energy. This is the important quantity in the present problem, and not so much the directional dependence of the radiance.



(a) Cost functional (b) L1-norm shape sensitivity



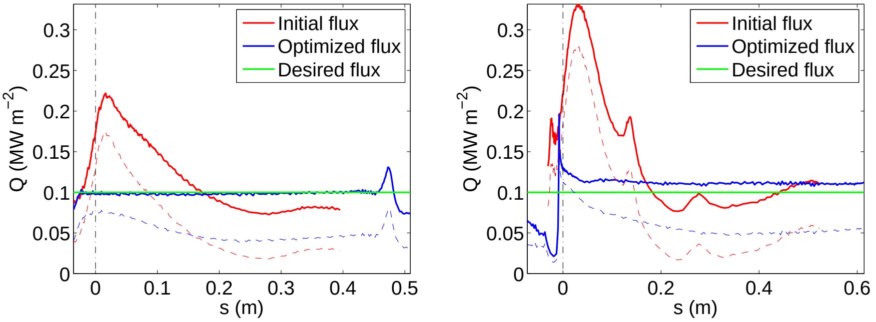
(c) Residuals state equations (d) Residuals adjoint equations

Figure 8.10: History of residuals during one-shot optimization.

Since the shape sensitivity is influenced by statistical fluctuations, it will never reach machine zero. As a result, the shape of the domain keeps changing, and the state and adjoint equations cannot truly converge. Therefore, as a practical criterion the optimization algorithm is stopped when the cost functional is not reduced significantly any more, after approximately 40.000 iterations in this case. Then, the state equations are allowed to converge. Compared to the 7.500 state iterations needed for a single state simulation, there is a factor of 10 in the computational cost of optimization due to plasma state and adjoint simulations only. On top of that, the forward and adjoint radiative transfer equations were solved 20 times each, which means a factor of 40 in cost for the radiation problem. Since the radiation problem takes 20% of the cost of a single plasma-and-radiation simulation, the total cost of one-shot optimization is equivalent to 16 coupled simulations. This is still a very powerful result.

Furthermore, the contribution of the radiation problem to the total cost may be reduced quite easily by parallelization of the Monte Carlo algorithm (for the test case, only 12 processors were used).

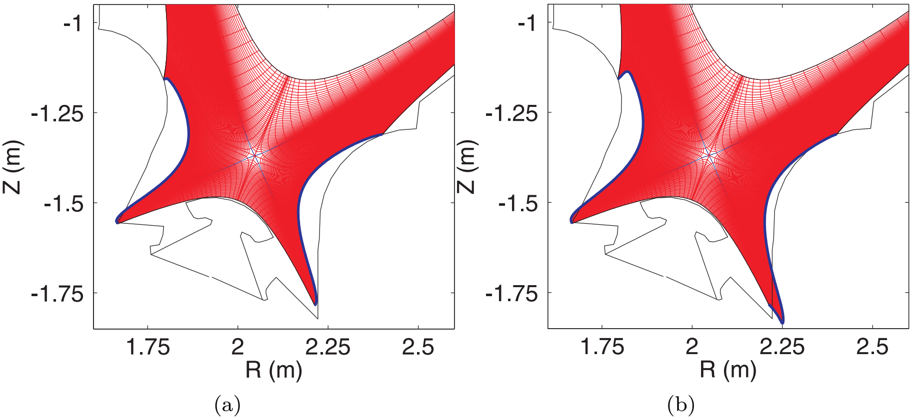
The optimized total energy fluxes are shown in Figure 8.11, and the corresponding optimized geometry in Fig. 8.12. At the inner target, an almost perfectly uniform load is obtained at the desired level. At the outer target, a uniform level is obtained over most of the target, but a bit higher than the desired level. Two numerical issues are noted. At the part of the inner target close to the main chamber wall, a surprising 'bump' has appeared in the target shape. This results in the strange behavior of the total energy flux in that area. It remains to be investigated whether this bump is an artefact of the fixed end point of the target, or whether inaccuracies in the shape sensitivities are at play. At the outer target, there appears to be a singularity in the flux close to the separatrix. Based on the very deep divertor leg with a strongly deformed grid, this is probably attributed to discretization errors. Further grid refinement in this area may provide an answer to this problem.



(a) Inner target (b) Outer target

Figure 8.11: Initial and optimized total target energy fluxes. Dashed lines represent the target energy without contribution from radiation.

Particle and energy balances of the optimized configurations are summarized in Table 8.2. For comparison, also the balances of the initial configuration and the configuration optimized in Chapter 5 are provided. Additionally, the values of total radiated power are included in the table. While optimization without taking into account the radiation load has lead to a regime of lower total recycling, now the recycling process is intensified. At the inner target, the level of the desired energy flux is reached by increasing the contributions coming from the plasma (Q_t and Q_sr). At the outer target, on the other hand, these contributions are reduced to accommodate the radiation.



(a)

(b)

Figure 8.12: Optimized divertor geometries, (a) without taking into account radiation (see Chapter 5) and (b) with radiation included in the cost functional.

Table 8.2: Comparison of balances at initial and optimized configurations. The middle column refers to the configuration optimized in Chapter 5 (see Table 5.2).

	Initial	Optimized	Optimized, rad.
Particle balance (10^{20} s^{-1})			
Flux from core	372.0	350.6	357.0
Flux to inner target	585.0	582.7	594.4
Flux to outer target	658.3	608.0	679.3
Flux to outer wall	119.2	122.5	120.8
Flux to PF	44.6	37.3	49.7
Volumetric source	1035.1	999.9	1087.2
Energy balance (MW)			
Flux from core	3.00	3.00	3.00
Flux to inner target	0.15	0.17	0.18
Flux to outer target	0.49	0.51	0.41
Flux to outer wall	0.09	0.09	0.09
Flux to PF	0.08	0.07	0.06
Neutral interactions (sink)	0.21	0.18	0.23
Impurity radiation (sink)	1.63	1.59	1.66
Other volumetric sources	-0.35	-0.39	-0.36
Total radiated power (MW)	1.82	1.77	1.86

8.6 Conclusion

In large-scale tokamak reactors and next-step machines, radiation can constitute a significant fraction of the energy load of plasma-facing components. Therefore, it should be included in divertor design from the start. In this chapter the adjoint methodology for shape optimization of coupled fluid-kinetic systems is extended to radiation modeling. The governing radiative transfer equation is introduced, and its adjoint is derived. The coupling of the (adjoint) radiation problem with the (adjoint) edge plasma equations is discussed. Where the forward radiation problem is solved in a postprocessing step, the adjoint radiation problem has to be computed in preprocessing. The sensitivities required for optimal shape design are derived and analyzed. Due to the intrinsic nature of the radiation problem, the directional dependence cannot be eliminated completely. Therefore, at least in principle the complete radiation distribution has to be known at the boundaries of the domain. However, with the assumption of isotropic radiation the sensitivity expressions can be simplified significantly. As a further advantage, the simplified expressions are less sensitive to statistical noise.

Numerical test problems including radiation only show that the one-shot algorithm can find the optimal target configurations. Gradient filtering and smoothing are essential to maintain the regularity of the domain.

Finally, the method developed in this chapter is applied to the design of divertor targets for uniform total energy load. As in the previous chapters, the power of the one-shot algorithm is demonstrated. The optimization problem is solved at a cost of approximately 16 plasma-radiation simulations.

Chapter 9

Conclusions

9.1 General Conclusions

Power and particle exhaust from tokamak reactors are critical aspects for the successful operation of a fusion power plant. Divertors have to be designed to handle the large steady-state power loads without exceeding limits imposed by the materials, while at the same time ensuring efficient Helium removal capabilities. Since the operating conditions of reactor-scale devices are inaccessible experimentally, tokamak reactor design strongly depends on numerical simulations with edge plasma codes as B2-EIRENE. These codes involve a multi-fluid description of the ions and electrons, coupled to a kinetic description of the neutrals. By detailed and consistent modeling of the relevant physical processes, they allow to extrapolate the current understanding of edge processes towards reactor-relevant parameters.

With the construction of ITER well on its way and the development of first DEMO reactor concepts, computational divertor design is a topic of high priority. Presently, edge codes as B2-EIRENE are used in a design-by-analysis approach, where divertor geometry and magnetic field are manually adjusted to meet design requirements. However, due to the complex nature of the edge plasma flows, the large number of design variables and different physics and engineering constraints, this method is computationally extremely demanding.

In computational aerodynamics, similar design challenges are effectively tackled through shape optimization methods. Design problems are formulated as mathematical optimization problems, where the shape is the control variable. The governing flow equations and boundary conditions, e.g. the Navier-Stokes equations, are introduced as constraints. The shape sensitivities required by the optimization algorithm are computed by an adjoint approach. In this way, the sensitivity with respect to all control variables can be computed at an equivalent cost of a single flow simulation, independent of the number of design variables. Furthermore, by using shape calculus and shape sensitivity analysis, shape sensitivities can be written as boundary integrals, and can therefore be computed at almost negligible computational cost. This is of particular importance for one-shot optimization algorithms, which solve the state, adjoint and design equations as one coupled system. Using these methods, the entire optimization problem can be solved at a cost of 5 to 10 forward flow simulations.

In this thesis, state-of-the-art methods from the field of aerodynamics are adapted for use on the tokamak power and particle exhaust problem, in order to achieve automated and efficient design strategies for divertors. Since the adjoint method is applied for the first time to edge plasma applications, attention is paid to the theoretical formulation of the optimization problems and the introduction of the necessary concepts. The theoretical framework is then applied to a series of somewhat simplified yet representative edge plasma models.

Divertor shape optimization for fluid edge plasma models

In the first part of the thesis, divertor optimization for fluid edge plasma models is considered. As a first contribution of this thesis, the adjoint equations are derived for a general set of convection-diffusion equations in orthogonal curvilinear coordinates. The resulting formulation is not only valid for the models used in this thesis, but is more generally applicable to the fluid models used in more complete edge codes. It is discussed how different cost functionals enter in the adjoint equations, and relevant design problems are proposed. Furthermore, the general structure of an edge plasma design code is elaborated. In particular, it is pointed out that using black-box or one-shot methods, the optimization code can be used with an existing edge code, and numerical routines already written for the edge code can be reused for the adjoint sensitivity code.

A thorough numerical validation of the shape sensitivities is performed. It is shown that the shape sensitivities computed with the continuous adjoint approach and shape sensitivity analysis agree well with the shape sensitivities obtained with finite differencing. In this respect, the importance of good grid quality is stressed. In order to avoid numerical oscillations in the grid, Sobolev smoothing of the shape sensitivity is introduced. Furthermore, the importance of using a nine-point discretization of fluxes on distorted grids is shown. On the one hand, neglecting the nine-point terms leads to loss of accuracy of the shape sensitivities. On the other hand, the nine-point stencil appears essential to capture the isotropic behavior of the neutrals on distorted grids. The latter reason is probably the most demanding reason to use nine-point stencils with fluid neutral models.

Three different optimization algorithms are compared in terms of computational performance: two gradient-based methods (a steepest descent and the BFGS method) and a one-shot algorithm. The one-shot method, which solves the state, adjoint, and design equations as one coupled system, emerges as the most efficient of the three. Optimization problems can be solved at a cost of only 4 flow simulations in simplified slab geometry, and only 10 simulations is realistic SOL geometry. This is consistent with results from the field of aerodynamics. In order to achieve this performance in realistic geometry, a grid morphing procedure is used to adapt grids to the changing target shapes. This makes the cost of the design update step negligible compared to the iterations of state and adjoint equations, and is essential for the performance of the one-shot algorithm.

In order to tackle the power exhaust problem, a cost functional is proposed which aims at spreading the total heat load as much as possible across the high-heat-flux plasma-facing components. The total heat flux contains contributions of energy convected and conducted by the plasma, and of potential energy released locally at the target due to surface recombination. Using this cost functional, divertors with almost perfectly uniform power loading are obtained, both in simplified and in realistic SOL geometry. Depending on the magnitude of the desired uniform load, the targets can even handle a larger total power yet with a significantly reduced peak load. A parameter scan around the design point shows that also at off-design situations the load is more uniform than the load in the initial configuration.

Divertor shape optimization for coupled fluid-kinetic edge plasma models

The second part of the thesis focuses on shape optimization for coupled fluid-kinetic systems. Both kinetic neutral models and radiative transfer problems are considered.

Shape optimization for edge plasma models including kinetic neutrals is studied for a 1D edge plasma model. The continuous adjoint method is used to derive an adjoint fluid-kinetic system, which in turn allows for the efficient computation of shape sensitivities. Since self-adjoint collision kernels are used in this thesis, the adjoint kinetic process is identical to the forward problem, so an identical Monte Carlo code can be used. Only sources and detector functions have to be adapted. Additionally, detector functions have to be implemented in the forward code to accumulate the derivatives of the plasma source terms with respect to the state variables. Unfortunately, while the complete 6D (5D) neutral distribution function is not needed in the forward problem, this full directional dependence naturally arises in the adjoint kinetic equations. In practical implementations, these terms will have to be approximated using reduced models.

Using a 1D edge plasma model, convergence and code speed issues can be investigated in detail. Specifically, the kinetic neutral model can also be solved with a finite volume method, allowing to decouple the study of these issues first without the additional complexity of statistical Monte Carlo noise. It is shown that depending on the recycling coefficient and boundary conditions, the coupled fluid-kinetic system may be stiff. As a result, the computational time required to solve the system significantly increases, even in the absence of statistical Monte Carlo noise. By using partially and fully coupled update equations rather than iterative solution schemes for the nonlinear system of equations, the computational cost could be reduced by a factor of 2 to 3.

When the kinetic neutral model is solved with a Monte Carlo code, the statistical noise appears in the residuals and poses serious challenges on the convergence of the coupled system. Different schemes to cope with this noise are studied. In cases where numerical errors due to statistical uncertainty are dominant, the uncorrelated-sampling regime appears as the most effective in reducing the uncertainty by simple time averaging of the solution. On the other hand, the time averages do not satisfy the governing equations, and global imbalances due to correlation terms which scale inversely proportional with the number of Monte Carlo histories remain. Using correlated sampling, convergence up to machine precision can be achieved despite the presence of statistical noise. Reduction of statistical noise can then be achieved with an ensemble averaging procedure, which additionally provides clear error estimates.

Finally, a significant speedup of the code can be achieved by using an advanced short-cycle model. The short-cycle model used in this thesis is found by integrating the kinetic equation over the velocity component, leading to typical fluid equations. However, instead of having to model or approximate the closure terms, these can now be computed directly with the Monte Carlo code. In this way, a short-cycle model fully consistent with the kinetic equation is obtained, which naturally allows to include volume recombination. Using this short-cycle model, a computational cost reduction of a factor 10 to 100 could be achieved.

Next, the 1D edge plasma model is used to study the influence of statistical noise on shape optimization. First, the adjoint shape sensitivities are investigated. They are computed both with correlated and uncorrelated sampling. Both methods are equally accurate, but again the uncorrelated-sampling method is more effective in averaging out statistical fluctuations. Due to partial cancellation of individual terms in the adjoint sensitivity, the statistical uncertainty on the sensitivity is relatively high. In order to put the shape sensitivities into perspective, they are compared to the shape sensitivities computed with finite differences and correlated sampling. Since the 1D Monte Carlo problem is ‘well resolved’ in the sense that every single particle feels the change in design variable, the statistical uncertainty is much smaller than with the continuous adjoint method. For this case, the use of discrete adjoint methods should be investigated.

When solving the shape optimization problem in the presence of statistical noise, the one-shot method has a very interesting advantage in combination with an uncorrelated-sampling method. Since relatively small design steps are required in the algorithm, these act as a smoothing filter for the design procedure. While the sign of a specific instance of the shape sensitivity may be wrong, on average the design still changes in the right direction. Depending on the level of the noise, the overall cost of the one-shot algorithm ranges from 20 to 100 flow simulations, with the largest factor corresponding to simulations with largest statistical uncertainty.

As a last application, a coupled 2D fluid-kinetic edge plasma model involving radiation transport is studied. In reactor-scale devices, the radiation constitutes a significant fraction of the power exhaust, and must therefore be taken into account in the design. First, the coupling of the plasma and radiation models is investigated. Under certain simplifying assumptions, the radiation problem can be solved in a post-processing step once the plasma solution has been obtained. In the adjoint problem, a corresponding adjoint radiative transfer equation is derived which has to be solved in preprocessing compared to the other adjoint equations. Shape sensitivities are derived for a cost functional which aims at spreading out the total target power load as much as possible, now including also the radiation load. Theoretical expressions are obtained, which unfortunately require complete knowledge of the radiation distribution. However, by assuming homogeneous radiation, the expressions can be simplified and the directional dependence is eliminated. Furthermore, the resulting expression are more robust with respect to statistical noise of the Monte Carlo radiation code.

To conclude the thesis, the divertor design optimization algorithm for uniform total power load is applied in realistic SOL geometry. The power of the one-shot algorithm is demonstrated again for this challenging test case.

9.2 Suggestions for Further Research

With the introduction of adjoint methods in edge plasma modeling, a wide range of possibilities and new applications emerge. As a direct continuation of the work, the design strategy developed in this thesis should be transferred to more complete edge codes. In order to achieve this, specific issues will have to be addressed. Some are of technical nature. For example, the use of nine-point discretization schemes and/or wide grids will be required, and it will have to be assessed whether current polynomial schemes used by the kinetic codes for interpolation of atomic and molecular data produce sufficiently accurate derivatives. On the other hand, extending the adjoint methods to a code as EIRENE will require a lot of dedicated theoretical work and careful programming. Also, the use of reliable, reduced neutral models will be essential for the computation of shape sensitivities without knowledge of the complete neutral distribution. Despite the challenges, extending the method will allow extrapolation of the methods to ITER-scale devices and beyond. While inevitably the computational cost will increase, these large-scale applications will allow to fully take advantage of the power of automated design.

Additionally, the design method can easily be extended to another critical divertor performance issue: Helium exhaust. As in the radiation problem, in a first approximation Helium transport can be modeled by a kinetic Monte Carlo code in a post-processing step. However, the computation of reliable shape sensitivities may be more challenging, since Helium transport can probably not be approximated as isotropic. Other design optimization goals may include minimization of sputtering and erosion, mitigating the effect of transient events as ELMs (Edge Localized Modes) and multiobjective optimization.

Furthermore, there are many exciting new applications of the adjoint sensitivity code. In general, an adjoint code allows to compute sensitivities with respect to all input parameters to the code. These include the geometry and magnetic field, but also model parameters (e.g. uncertain radial transport coefficients) and boundary conditions. The application to magnetic field optimization has already been initiated by M. Blommaert [11], and shows great potential. J. De Schutter [23] has shown that the method can be applied for the efficient estimation of unknown model parameters, for example based on experimentally measured profiles. Another research track that holds much promise is the use of adjoint methods for robust design. Also an emerging field in aerodynamic design, see e.g. Refs. [111, 112], these design methods strive to be robust with respect to variations in the operating conditions of the reactor, as well as geometrical uncertainties.

Other future research tracks should concentrate more on numerical issues, convergence, and code speed. While the adjoint-based optimization methods significantly reduce computational cost in terms of number of simulations, the computational time required for a single simulation will remain a critical bottleneck. Based on the results of convergence studies for coupled fluid-kinetic codes, some interesting iterative solution schemes and coupling methods for fluid-kinetic codes have been identified. First, an important code speedup may be achieved by using more advanced iterative schemes for the coupled fluid-kinetic equations. Furthermore, it should be investigated whether correlated-sampling techniques combined with advanced short-cycle models can provide the same significant speedup in a code as B2-EIRENE and at reactor-relevant parameters. On the other hand, reduction of statistical noise through uncorrelated sampling, possibly in combination with the Robbins-Monro method from stochastic approximation [100] could be subject of further research.

In this respect, it would be very interesting to investigate whether adjoint methods can contribute to code speedup, for example by automatically selecting the optimal (local) time steps and possibly other parameters in the fluid-kinetic coupling schemes. Other applications in this regard include optimal, automated gridding for the minimization of discretization errors.

Appendix A

Tangential Calculus and Shape Sensitivity Analysis

In this appendix, some useful expressions from the field of tangential calculus and shape sensitivity analysis are summarized. These expressions are a collection of results from the literature, but since they form the basis for the computation of shape sensitivities they are provided here for reference. First, in Section A.1 some general definitions and identities of tangential calculus are given. Then, Section A.2 studies the behavior of geometric quantities under a transformation of the domain. The presentation of results in this appendix is formal. A thorough discussion of these topics can be found in for example Refs. [114, 52, 19, 110, 29].

A.1 Tangential Calculus

The discussion in this section is based mainly on Ref. [29]. For a d-dimensional domain Ω with boundary Σ, consider the (local) parametrization $h : B \to U$ from the unit ball B to U, see Fig. A.1, so a point $\boldsymbol{\omega}$ is given by $\boldsymbol{\omega} = h(\xi_1, ..., \xi_d)$. It is assumed that the parametrization of the domain is chosen so its boundary is determined by $\xi_d = 0$, the interior by $\xi_d > 0$ and the exterior by $\xi_d < 0$. For ease of notation, the boundary will be denoted by $h(\xi, 0)$. The basis vectors in the plane tangent to the boundary are determined by the Jacobian matrix of the parametrization,

$$\boldsymbol{\tau}_i = Dh(\xi, 0)\mathbf{e}_i, \quad i = 1, ..., d-1, \tag{A.1}$$

where $\mathbf{e}_i$ are the unit vectors in $\mathbb{R}^d$. Note that the vectors $\boldsymbol{\tau}_i$ are not necessarily unit vectors, and not necessarily orthogonal to each other. The outward pointing unit normal to the plane is

$$\boldsymbol{\nu} = -\frac{Dh(\xi,0)^{-T}\mathbf{e}_d}{||Dh(\xi,0)^{-T}\mathbf{e}_d||}. \tag{A.2}$$

Indeed, it can be verified by direction computation that for $i = 1, ..., d-1$, $\boldsymbol{\nu} \cdot \boldsymbol{\tau}_i = 0$.

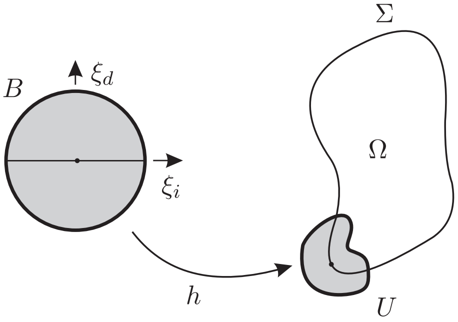


Figure A.1: Local parametrization of the domain. Figure based on Ref. [29].

The directional derivative of a (scalar of vector valued) function f in direction $\mathbf{d}$ is defined as

$$\frac{\partial f}{\partial \mathbf{d}} = Df\mathbf{d} = \mathbf{d} \cdot \nabla f = \nabla_{\mathbf{d}} f. \tag{A.3}$$

The last notation is used most frequently in the text. Specifically, the derivative in the normal direction is

$$\frac{\partial f}{\partial \boldsymbol{\nu}} = \boldsymbol{\nu} \cdot \nabla f = \nabla_{\boldsymbol{\nu}} f \tag{A.4}$$

and the component of the gradient in the normal direction is

$$\frac{\partial f}{\partial \boldsymbol{\nu}}\boldsymbol{\nu} = (\boldsymbol{\nu} \cdot \nabla f)\boldsymbol{\nu} = (\nabla_{\boldsymbol{\nu}} f)\boldsymbol{\nu}. \tag{A.5}$$

The tangential gradient of a function f is defined as the orthogonal projection of its gradient onto the tangent plane,

$$\nabla_\Sigma f \equiv \nabla f|_\Sigma - \frac{\partial f}{\partial \boldsymbol{\nu}}\boldsymbol{\nu}. \tag{A.6}$$

The tangential Jacobian matrix of a vector function $\mathbf{v} = (v_1, ..., v_M)^T$ is defined using the tangential gradient from Eq. (A.6):

$$D_\Sigma \mathbf{v} = (\nabla_\Sigma v_1, ..., \nabla_\Sigma v_M)^T = D\mathbf{v}|_\Sigma - D\mathbf{v}\boldsymbol{\nu}\boldsymbol{\nu}^T, \tag{A.7}$$

With this definition, the tangential divergence becomes

$$\nabla_\Sigma \cdot \mathbf{v} = \mathrm{tr}(D_\Sigma \mathbf{v}) = \nabla \cdot \mathbf{v}|_\Sigma - \boldsymbol{\nu} \cdot \nabla \mathbf{v} \cdot \boldsymbol{\nu}. \tag{A.8}$$

In case the vectors $\boldsymbol{\tau}_i$ form an orthonormal basis for the tangent plane, the tangential gradient and divergence can easily be written in terms of the directional derivatives. For the tangential gradient, the expression

$$\nabla_\Sigma f = \sum_{i=1}^{d-1} \frac{\partial f}{\partial \boldsymbol{\tau}_i} \boldsymbol{\tau}_i \tag{A.9}$$

is obtained. The tangential divergence becomes

$$\nabla_\Sigma \cdot \mathbf{v} = \sum_{i=1}^{d-1} \frac{\partial \mathbf{v}}{\partial \boldsymbol{\tau}_i} \cdot \boldsymbol{\tau}_i = \sum_{i=1}^{d-1} \boldsymbol{\tau}_i \cdot \nabla \mathbf{v} \cdot \boldsymbol{\tau}_i. \tag{A.10}$$

The typical identity $\nabla \cdot (f\mathbf{v}) = f\,\nabla \cdot \mathbf{v} + \mathbf{v} \cdot \nabla f$, with f a scalar valued function f and $\mathbf{v}$ a vector also holds in the tangent plane:

$$\nabla_\Sigma \cdot (f\mathbf{v}) = f\,\nabla_\Sigma \cdot \mathbf{v} + \mathbf{v} \cdot \nabla_\Sigma f. \tag{A.11}$$

Other useful identities are the tangential Stokes formula,

$$\int_\Sigma \nabla_\Sigma \cdot \mathbf{v}\,\mathrm{d}\sigma = \int_\Sigma \kappa \mathbf{v} \cdot \boldsymbol{\nu}\,\mathrm{d}\sigma, \tag{A.12}$$

and the tangential Green's formula, obtained by combining Eqs. (A.11) and (A.12):

$$\int_\Sigma f\,\nabla_\Sigma \cdot \mathbf{v} + \mathbf{v} \cdot \nabla_\Sigma f\,\mathrm{d}\sigma = \int_\Sigma \kappa f \mathbf{v} \cdot \boldsymbol{\nu}\,\mathrm{d}\sigma. \tag{A.13}$$

By definition, the additive curvature is the sum of the principal curvatures of the boundary. It can be computed from the outward unit normal field by

$$\kappa = \nabla_\Sigma \cdot \boldsymbol{\nu}. \tag{A.14}$$

A.2 Shape Sensitivity Analysis

In this section, the shape and material derivatives of some frequently appearing quantities are studied. Specifically, some interesting relations regarding the transformation of volumes, surface elements and the unit normal field are given. More details can be found in the references given in the introduction to this appendix, specifically in Refs. [19, 29]. For ease of presentation, it is assumed that the transformation $\mathbf{T}_t$ between the unperturbed and perturbed domains due to a velocity field $\boldsymbol{\mathcal{V}}(\boldsymbol{\omega})$ is given by the perturbation of identity,

$$\mathbf{T}_t(\boldsymbol{\omega}) = \boldsymbol{\omega} + t\boldsymbol{\mathcal{V}}(\boldsymbol{\omega}). \tag{A.15}$$

Material derivative of Jacobian matrix

For almost all results below, the Jacobian matrix comes into play. For the perturbation of identity, this Jacobian matrix is

$$D\mathbf{T}_t = I + tD\boldsymbol{\mathcal{V}}(\boldsymbol{\omega}) = I + t(\nabla\boldsymbol{\mathcal{V}}(\boldsymbol{\omega}))^T, \tag{A.16}$$

with I the identity. At $t = 0$ the Jacobian matrix is $D\mathbf{T}_0 = I$ as it should. The material derivative of the Jacobian matrix is

$$\left.\frac{\mathrm{d}}{\mathrm{d}t} D\mathbf{T}_t\right|_{t=0+} = D\boldsymbol{\mathcal{V}}(\boldsymbol{\omega}), \tag{A.17}$$

and using

$$\begin{aligned} 0 = \left.\frac{\mathrm{d}I}{\mathrm{d}t}\right|_{t=0+} &= \left.\frac{\mathrm{d}}{\mathrm{d}t}(D\mathbf{T}_t D\mathbf{T}_t^{-1})\right|_{t=0+} \\ &= \left.\frac{\mathrm{d}}{\mathrm{d}t}(D\mathbf{T}_t) D\mathbf{T}_t^{-1}\right|_{t=0+} + \left. D\mathbf{T}_t \frac{\mathrm{d}}{\mathrm{d}t} D\mathbf{T}_t^{-1}\right|_{t=0+} \end{aligned}$$

the material derivative of the inverse transformation is

$$\left.\frac{\mathrm{d}}{\mathrm{d}t} D\mathbf{T}_t^{-1}\right|_{t=0+} = -\left.\frac{\mathrm{d}}{\mathrm{d}t} D\mathbf{T}_t\right|_{t=0+} = -D\boldsymbol{\mathcal{V}}(\boldsymbol{\omega}). \tag{A.18}$$

Material derivative of volume element

The transformation of a volume element is determined by the Jacobian, i.e. the determinant of the Jacobian matrix: $\mathrm{d}\boldsymbol{\omega}_t = |D\mathbf{T}_t|\,\mathrm{d}\boldsymbol{\omega}$. For meaningful transformations, the Jacobian is positive (otherwise, negative volumes would be produced). The material derivative of the Jacobian can be found by direct computation as

$$\left.\frac{\mathrm{d}}{\mathrm{d}t} |D\mathbf{T}_t|\right|_{t=0+} = \nabla \cdot \boldsymbol{\mathcal{V}}. \tag{A.19}$$

Material derivative of surface normal

The material derivative of the surface normal can be computed conveniently by considering again the local parametrization h in a composition of maps. The location of the perturbed boundary is then $\Sigma_t = \mathbf{T}_t(\Sigma) = \mathbf{T}_t(h(\xi, 0))$. Therefore, using expression (A.2) and the chain rule, the normal on the perturbed domain is

$$\boldsymbol{\nu}_t = -\frac{D\mathbf{T}_t^{-T} Dh(\xi,0)^{-T}\mathbf{e}_d}{\left|\left|D\mathbf{T}_t^{-T} Dh(\xi,0)^{-T}\mathbf{e}_d\right|\right|} = \frac{D\mathbf{T}_t^{-T}\boldsymbol{\nu}}{\left|\left|D\mathbf{T}_t^{-T}\boldsymbol{\nu}\right|\right|}. \tag{A.20}$$

The material derivative of the normal is thus

$$\begin{aligned} \left.\frac{\mathrm{d}\boldsymbol{\nu}_t}{\mathrm{d}t}\right|_{t=0+} &= \left.\frac{\mathrm{d}}{\mathrm{d}t}\left(D\mathbf{T}_t^{-T}\boldsymbol{\nu}\right)\right|_{t=0+} - \boldsymbol{\nu}\left(\boldsymbol{\nu}\cdot\left.\frac{\mathrm{d}}{\mathrm{d}t}\left(D\mathbf{T}_t^{-T}\boldsymbol{\nu}\right)\right|_{t=0+}\right) \\ &= -\nabla\boldsymbol{\mathcal{V}}\cdot\boldsymbol{\nu} + \boldsymbol{\nu}(\boldsymbol{\nu}\cdot\nabla\boldsymbol{\mathcal{V}}\cdot\boldsymbol{\nu}). \end{aligned} \tag{A.21}$$

Using expression (A.7), this simplifies to

$$\left.\frac{\mathrm{d}\boldsymbol{\nu}_t}{\mathrm{d}t}\right|_{t=0+} = -\nabla_\Sigma\boldsymbol{\mathcal{V}}\cdot\boldsymbol{\nu}. \tag{A.22}$$

Material derivative of surface element

A differential surface element on the transformed domain is given by

$$\boldsymbol{\nu}_t \mathrm{d}\sigma_t = |D\mathbf{T}_t|\, D\mathbf{T}_t^{-T}\boldsymbol{\nu}\,\mathrm{d}\sigma, \tag{A.23}$$

so the size of the element transforms as

$$\mathrm{d}\sigma_t = |D\mathbf{T}_t|\,\left|\left|D\mathbf{T}_t^{-T}\boldsymbol{\nu}\right|\right|\,\mathrm{d}\sigma. \tag{A.24}$$

Thus, in order to describe the change in area, the material derivative of $|D\mathbf{T}_t|\,\left|\left|D\mathbf{T}_t^{-T}\boldsymbol{\nu}\right|\right|$ is needed. Elaboration using (A.19) and (A.18) gives

$$\left.\frac{\mathrm{d}}{\mathrm{d}t}\,|D\mathbf{T}_t|\,\left|\left|D\mathbf{T}_t^{-T}\boldsymbol{\nu}\right|\right|\right|_{t=0+} = \nabla\cdot\boldsymbol{\mathcal{V}} - \boldsymbol{\nu}\cdot\nabla\boldsymbol{\mathcal{V}}\cdot\boldsymbol{\nu}. \tag{A.25}$$

Material derivative of surface integrals

All elements are now at hand to derive the shape derivative given in Eq. (2.20). Using Eq. (A.24) gives

$$\begin{aligned} \dot{J}_1(\Omega;\boldsymbol{\mathcal{V}}) &= \left.\frac{\mathrm{d}}{\mathrm{d}t}\left(\int_{\Sigma_t} g\left(t,\boldsymbol{\omega}_t\right)\,\mathrm{d}\sigma_t\right)\right|_{t=0+} \\ &= \int_\Sigma \left.\left(g\left(t,\boldsymbol{\omega}_t\right)|D\mathbf{T}_t|\,\left|\left|D\mathbf{T}_t^{-T}\boldsymbol{\nu}\right|\right|\right)\right|_{t=0+}\,\mathrm{d}\sigma. \end{aligned}$$

Next, using the definition of the shape derivative (2.14) and Eq. (A.25) leads to

$$\begin{aligned}\dot{J}_1(\Omega;\boldsymbol{\mathcal{V}}) &= \int_\Sigma \left(g' + \boldsymbol{\mathcal{V}}\cdot\nabla g + g\left(\nabla\cdot\boldsymbol{\mathcal{V}} - \boldsymbol{\nu}\cdot\nabla\boldsymbol{\mathcal{V}}\cdot\boldsymbol{\nu}\right)\right)\,\mathrm{d}\sigma \\ &= \int_\Sigma \left(g' + \boldsymbol{\mathcal{V}}\cdot\nabla g + g\nabla_\Sigma\cdot\boldsymbol{\mathcal{V}}\right)\,\mathrm{d}\sigma.\end{aligned}$$

For the last equality, the definition of the tangential divergence (A.8) has been used. The result (2.23) is then found by applying the tangential Green's formula, Eq. (A.13), and the definition of the tangential gradient (A.6),

$$\dot{J}_1(\Omega;\boldsymbol{\mathcal{V}}) = \int_\Sigma g'\,\mathrm{d}\sigma + \int_\Sigma \left(\boldsymbol{\nu}\cdot\nabla g + g\kappa\right)\boldsymbol{\mathcal{V}}\cdot\boldsymbol{\nu}\,\mathrm{d}\sigma. \tag{A.26}$$

The shape derivatives of the other two boundary integrals are obtained in a similar way. For further elaboration of Eq. (2.21), see Ref. [19], and for (2.22), see Refs. [108, 110].

Appendix B

Adjoint Equations in Poloidal-Radial Coordinate System

In this appendix, the adjoint equations for the edge plasma model described in Section 3.1 are elaborated. For easy reference, the general convection-diffusion form of the state equations, Eq. (3.35), is repeated here:

$$\begin{aligned} 0 \;=\; \boldsymbol{\mathcal{B}}(\mathbf{q}) \;=\; & S(\mathbf{q}, \nabla_\theta \mathbf{q}, \nabla_r \mathbf{q}) - \frac{1}{\sqrt{g}} \frac{\partial}{\partial \theta} \left(\frac{\sqrt{g}}{h_\theta} C^\theta(\mathbf{q}) - \frac{\sqrt{g}}{h_\theta^2} D^\theta(\mathbf{q}) \frac{\partial \mathbf{q}}{\partial \theta} \right) \\ & - \frac{1}{\sqrt{g}} \frac{\partial}{\partial r} \left(\frac{\sqrt{g}}{h_r} C^r(\mathbf{q}) - \frac{\sqrt{g}}{h_r^2} D^r(\mathbf{q}) \frac{\partial \mathbf{q}}{\partial r} \right). \end{aligned} \tag{B.1}$$

The vector of state variables is $\mathbf{q} = (n, u_{||}, p_\mathrm{n}, T)^T$. $C^\theta(\mathbf{q})$ and $C^r(\mathbf{q})$ are the poloidal and radial convective fluxes,

$$C^\theta(\mathbf{q}) \;=\; \begin{pmatrix} n u_\theta \\ m n u_\theta u_{||} \\ n_{\mathrm{n,eq}} u_\theta \\ \frac{5}{2}((1+Z_\mathrm{i}) n u_\theta + n_{\mathrm{n,eq}} u_\theta) T \end{pmatrix}, \quad C^r(\mathbf{q}) \;=\; \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix},$$

while $D^\theta(\mathbf{q})$ and $D^r(\mathbf{q})$ represent generalized diffusion matrices,

$$\begin{aligned} D^\theta(\mathbf{q}) &= \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & \eta^{\mathrm{i}}_\theta & 0 & 0 \\ 0 & 0 & D^{\mathrm{n}}_p & 0 \\ 0 & 0 & \frac{5}{2}TD^{\mathrm{n}}_p & \kappa_\theta \end{pmatrix}, \\ D^r(\mathbf{q}) &= \begin{pmatrix} D^{\mathrm{i}} & 0 & 0 & 0 \\ mu_{||}D^{\mathrm{i}} & \eta^{\mathrm{i}}_r & 0 & 0 \\ 0 & 0 & D^{\mathrm{n}}_p & 0 \\ \frac{5}{2}(1+Z_{\mathrm{i}})TD^{\mathrm{i}} & 0 & \frac{5}{2}TD^{\mathrm{n}}_p & \kappa_r \end{pmatrix}. \end{aligned}$$

The source terms in the equations are

$$S(\mathbf{q},\nabla_\theta\mathbf{q},\nabla_r\mathbf{q}) = S_{\mathrm{n}}(\mathbf{q},\nabla_\theta\mathbf{q},\nabla_r\mathbf{q}) + S_z(\mathbf{q},\nabla_\theta\mathbf{q},\nabla_r\mathbf{q}) + S_p(\mathbf{q},\nabla_\theta\mathbf{q},\nabla_r\mathbf{q}),$$

with

$$\begin{aligned} S_{\mathrm{n}} &= \begin{pmatrix} n_{\mathrm{e}}n_{\mathrm{n}}K_{\mathrm{i}} - n_{\mathrm{i}}n_{\mathrm{e}}K_{\mathrm{r}} \\ mn_{\mathrm{e}}n_{\mathrm{n}}K_{\mathrm{i}}u_{\mathrm{n}||} - mn_{\mathrm{i}}n_{\mathrm{e}}K_{\mathrm{r}}u_{||} - mn_{\mathrm{i}}n_{\mathrm{n}}K_{\mathrm{cx}}(u_{||} - u_{\mathrm{n}||}) \\ n_{\mathrm{i}}n_{\mathrm{e}}K_{\mathrm{r}} - n_{\mathrm{e}}n_{\mathrm{n}}K_{\mathrm{i}} \\ S_{E_{\mathrm{c}}} - E_{\mathrm{i}}n_{\mathrm{e}}n_{\mathrm{n}}K_{\mathrm{i}} \end{pmatrix}, \\ S_z &= \begin{pmatrix} 0 \\ 0 \\ 0 \\ -c_z n_{\mathrm{i}} n_{\mathrm{e}} L_z \end{pmatrix}, \\ S_p &= \begin{pmatrix} 0 \\ -\frac{b_\theta}{h_\theta}\frac{\partial p}{\partial\theta} \\ 0 \\ \eta^{\mathrm{i}}_\theta\left(\frac{1}{h_\theta}\frac{\partial u_{||}}{\partial\theta}\right)^2 + \eta^{\mathrm{i}}_r\left(\frac{1}{h_r}\frac{\partial u_{||}}{\partial r}\right)^2 + \dots \\ \dots + u_{||}\frac{b_\theta}{h_\theta}\frac{\partial p}{\partial\theta} + u_{\mathrm{n}\theta}\frac{1}{h_\theta}\frac{\partial p_{\mathrm{n}}}{\partial\theta} + u_{\mathrm{n}r}\frac{1}{h_r}\frac{\partial p_{\mathrm{n}}}{\partial r} \end{pmatrix}. \end{aligned} \tag{B.2}$$

In the corresponding adjoint field equations and boundary conditions, linearizations of the model matrices with respect to the state variables and their gradients are needed. Linearizing the poloidal convective fluxes leads to

$$(C^\theta_{\mathbf{q}})^T = \begin{pmatrix} u_\theta & mu_\theta u_{||} & \frac{\partial n_{\mathrm{n,eq}}}{\partial n}u_\theta & \frac{5}{2}\left(1 + Z_{\mathrm{i}} + \frac{\partial n_{\mathrm{n,eq}}}{\partial n}\right)u_\theta T \\ nb_\theta & 2mnu_\theta & n_{\mathrm{n,eq}}b_\theta & \frac{5}{2}\left((1+Z_{\mathrm{i}})n + n_{\mathrm{n,eq}}\right)b_\theta T \\ 0 & 0 & \frac{\partial n_{\mathrm{n,eq}}}{\partial p_{\mathrm{n}}}u_\theta & \frac{5}{2}\frac{\partial n_{\mathrm{n,eq}}}{\partial p_{\mathrm{n}}}u_\theta T \\ 0 & 0 & \frac{\partial n_{\mathrm{n,eq}}}{\partial T}u_\theta & \frac{5}{2}\left((1+Z_{\mathrm{i}})n + n_{\mathrm{n,eq}} + \frac{\partial n_{\mathrm{n,eq}}}{\partial T}T\right)u_\theta \end{pmatrix}.$$

The linearized diffusion matrices are

$$D_n^\theta = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{\partial D_p^\mathrm{n}}{\partial n} & 0 \\ 0 & 0 & \frac{5}{2}T\frac{\partial D_p^\mathrm{n}}{\partial n} & \frac{\partial \kappa_\theta}{\partial n} \end{pmatrix}, \qquad D_{u_{||}}^\theta = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix},$$

$$D_T^\theta = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & \frac{\partial \eta_\theta^\mathrm{i}}{\partial T} & 0 & 0 \\ 0 & 0 & \frac{\partial D_p^\mathrm{n}}{\partial T} & 0 \\ 0 & 0 & \frac{5}{2}\left(D_p^\mathrm{n} + T\frac{\partial D_p^\mathrm{n}}{\partial T}\right) & \frac{\partial \kappa_\theta}{\partial T} \end{pmatrix}, \quad D_{p_\mathrm{n}}^\theta = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{\partial \kappa_\theta}{\partial p_\mathrm{n}} \end{pmatrix},$$

and

$$D_n^r = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & \frac{\partial \eta_r^\mathrm{i}}{\partial n} & 0 & 0 \\ 0 & 0 & \frac{\partial D_p^\mathrm{n}}{\partial n} & 0 \\ 0 & 0 & \frac{5}{2}T\frac{\partial D_p^\mathrm{n}}{\partial n} & \frac{\partial \kappa_r}{\partial n} \end{pmatrix}, \qquad D_{u_{||}}^r = \begin{pmatrix} 0 & 0 & 0 & 0 \\ mD^\mathrm{i} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix},$$

$$D_T^r = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{\partial D_p^\mathrm{n}}{\partial T} & 0 \\ \frac{5}{2}(1+Z_\mathrm{i})D^\mathrm{i} & 0 & \frac{5}{2}\left(D_p^\mathrm{n}+T\frac{\partial D_p^\mathrm{n}}{\partial T}\right) & \frac{\partial \kappa_r}{\partial T} \end{pmatrix}, \quad D_{p_\mathrm{n}}^r = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{\partial \kappa_r}{\partial p_\mathrm{n}} \end{pmatrix}.$$

Finally, the source terms in the state equations have to be linearized with respect to gradients in the state variables,

$$S_{\nabla_\theta \mathbf{q}}^T = \begin{pmatrix} 0 & -2b_\theta T & 0 & 2u_\theta T \\ 0 & 0 & 0 & 2\eta_\theta^\mathrm{i}\frac{1}{h_\theta}\frac{\partial u_{||}}{\partial \theta} \\ 0 & -mn_\mathrm{n}(n_\mathrm{e}K_\mathrm{i}+n_\mathrm{i}K_\mathrm{cx})b_\theta\frac{D_p^\mathrm{n}}{n_\mathrm{n}} & 0 & u_{\mathrm{n}\theta}-\frac{D_p^\mathrm{n}}{n_\mathrm{n}}\frac{1}{h_\theta}\frac{\partial p_\mathrm{n}}{\partial \theta}+2mK\frac{D_p^\mathrm{n}}{n_\mathrm{n}}(u_\theta-u_{\mathrm{n}\theta}) \\ 0 & -2b_\theta n & 0 & 2u_\theta n \end{pmatrix},$$

$$S_{\nabla_r \mathbf{q}}^T = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2\eta_r^\mathrm{i}\frac{1}{h_r}\frac{\partial u_{||}}{\partial r} \\ 0 & 0 & 0 & 2\left(1-mK\frac{D_p^\mathrm{n}}{n_\mathrm{n}}\right)u_{\mathrm{n}r} \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

The linearization of the source terms with respect to **q** will be done in Section B.1.

With these matrices at hand, the adjoint equations and boundary conditions can now be elaborated.

B.1 Derivation Adjoint Field Equations

According to the derivation in Chapter 3, Section 3.3, the adjoint field equations are given by Eq. (3.50),

$$\nabla_{\mathbf{q}} J_1(\Omega, \mathbf{q}) + \boldsymbol{\mathcal{B}}^*_{\mathbf{q}}(\mathbf{q})\mathbf{q}^* = 0 \quad \text{in} \quad \Omega, \tag{B.3}$$

where the operator $\boldsymbol{\mathcal{B}}^*_{\mathbf{q}}\mathbf{q}^*$ in conservative form is (see Eq. (3.49))

$$\begin{aligned}
\boldsymbol{\mathcal{B}}^*_{\mathbf{q}}\mathbf{q}^* \quad = \quad & S^T_{\mathbf{q}}\mathbf{q}^* - \frac{1}{\sqrt{g}}\frac{\partial}{\partial\theta}\left(\frac{\sqrt{g}}{h_\theta}S^T_{\nabla_\theta\mathbf{q}}\mathbf{q}^*\right) - \frac{1}{\sqrt{g}}\frac{\partial}{\partial r}\left(\frac{\sqrt{g}}{h_r}S^T_{\nabla_r\mathbf{q}}\mathbf{q}^*\right) \\
& + \frac{1}{\sqrt{g}}\frac{\partial}{\partial\theta}\left(\frac{\sqrt{g}}{h_\theta}\left(C^\theta_{\mathbf{q}}\right)^T\mathbf{q}^* + \frac{\sqrt{g}}{h_\theta^2}(D^\theta)^T\frac{\partial\mathbf{q}^*}{\partial\theta}\right) \\
& + \frac{1}{\sqrt{g}}\frac{\partial}{\partial r}\left(\frac{\sqrt{g}}{h_r}\left(C^r_{\mathbf{q}}\right)^T\mathbf{q}^* + \frac{\sqrt{g}}{h_r^2}(D^r)^T\frac{\partial\mathbf{q}^*}{\partial r}\right) \\
& - \frac{1}{\sqrt{g}}\frac{\partial}{\partial\theta}\left(\frac{\sqrt{g}}{h_\theta}\left(C^\theta_{\mathbf{q}}\right)^T\right)\mathbf{q}^* - \frac{1}{\sqrt{g}}\frac{\partial}{\partial r}\left(\frac{\sqrt{g}}{h_r}\left(C^r_{\mathbf{q}}\right)^T\right)\mathbf{q}^* \\
& - \frac{1}{h_\theta}\frac{\partial(\mathbf{q}^*)^T}{\partial\theta}D^\theta_{\mathbf{q}}\frac{1}{h_\theta}\frac{\partial\mathbf{q}}{\partial\theta} - \frac{1}{h_r}\frac{\partial(\mathbf{q}^*)^T}{\partial r}D^r_{\mathbf{q}}\frac{1}{h_r}\frac{\partial\mathbf{q}}{\partial r}.
\end{aligned} \tag{B.4}$$

By decoupling the adjoint operator (B.4) into its four components and regrouping some terms, the individual adjoint equations are found. These are the adjoint continuity equation,

$$\begin{aligned}
&-\frac{1}{\sqrt{g}}\frac{\partial}{\partial \theta}\left(\frac{\sqrt{g}}{h_\theta}\left(u_\theta n^* + (m u_\theta u_{||} + 2 b_\theta T) u_{||}^* + \frac{\partial n_{\mathrm{n,eq}}}{\partial n} u_\theta p_{\mathrm{n}}^*\right)\right)\\
&-\frac{1}{\sqrt{g}}\frac{\partial}{\partial \theta}\left(\frac{\sqrt{g}}{h_\theta}\left(\frac{5}{2}\left(1 + Z_{\mathrm{i}} + \frac{\partial n_{\mathrm{n,eq}}}{\partial n}\right) - 2\right) u_\theta T T^*\right)\\
&-\frac{1}{\sqrt{g}}\frac{\partial}{\partial r}\left(\frac{\sqrt{g}}{h_r^2}\left(D^{\mathrm{i}}\frac{\partial n^*}{\partial r} + m D^{\mathrm{i}} u_{||}\frac{\partial u_{||}^*}{\partial r} + \frac{5}{2}\left(1 + Z_{\mathrm{i}}\right) D^{\mathrm{i}} T \frac{\partial T^*}{\partial r}\right)\right)\\
&= S_{n_{\mathrm{i}}}^* - n^* \frac{1}{\sqrt{g}}\frac{\partial}{\partial \theta}\left(\frac{\sqrt{g}}{h_\theta} u_\theta\right) - u_{||}^* \frac{1}{\sqrt{g}}\frac{\partial}{\partial \theta}\left(\frac{\sqrt{g}}{h_\theta} m u_\theta u_{||}\right) - 2 b_\theta u_{||}^* \frac{1}{h_\theta}\frac{\partial T}{\partial \theta}\\
&\qquad - p_{\mathrm{n}}^* \frac{1}{\sqrt{g}}\frac{\partial}{\partial \theta}\left(\frac{\sqrt{g}}{h_\theta}\frac{\partial n_{\mathrm{n,eq}}}{\partial n} u_\theta\right) - T^* \frac{1}{\sqrt{g}}\frac{\partial}{\partial \theta}\left(\frac{\sqrt{g}}{h_\theta}\frac{5}{2}\left(1 + Z_{\mathrm{i}}\right) u_\theta T\right)\\
&\qquad - T^* \frac{1}{\sqrt{g}}\frac{\partial}{\partial \theta}\left(\frac{\sqrt{g}}{h_\theta}\frac{5}{2}\frac{\partial n_{\mathrm{n,eq}}}{\partial n} u_\theta T\right) + 2 u_\theta T^* \frac{1}{h_\theta}\frac{\partial T}{\partial \theta}\\
&\qquad + T^* \frac{\partial n_{\mathrm{n,eq}}}{\partial n}\frac{u_\theta}{n_{\mathrm{n}}}\frac{1}{h_\theta}\frac{\partial p_{\mathrm{n}}}{\partial \theta} + \frac{\partial J_1}{\partial n},
\end{aligned}\tag{B.5}$$

the adjoint momentum equation,

$$\begin{aligned}
&-\frac{1}{\sqrt{g}}\frac{\partial}{\partial \theta}\left(\frac{\sqrt{g}}{h_\theta} 2 m n u_\theta u_{||}^* + \frac{\sqrt{g}}{h_\theta^2}\eta_\theta^{\mathrm{i}}\frac{\partial u_{||}^*}{\partial \theta}\right) - \frac{1}{\sqrt{g}}\frac{\partial}{\partial r}\left(\frac{\sqrt{g}}{h_r} m n u_r u_{||}^* + \frac{\sqrt{g}}{h_r^2}\eta_r^{\mathrm{i}}\frac{\partial u_{||}^*}{\partial r}\right)\\
&-\frac{1}{\sqrt{g}}\frac{\partial}{\partial \theta}\left(\frac{\sqrt{g}}{h_\theta}\left(n b_\theta n^* + n_{\mathrm{n,eq}} b_\theta p_{\mathrm{n}}^* + \frac{5}{2}\left((1 + Z_{\mathrm{i}}) n + n_{\mathrm{n,eq}}\right) b_\theta T T^*\right)\right)\\
&+\frac{1}{\sqrt{g}}\frac{\partial}{\partial \theta}\left(\frac{\sqrt{g}}{h_\theta^2} 2 \eta_\theta^{\mathrm{i}} T^* \frac{\partial u_{||}}{\partial \theta}\right) + \frac{1}{\sqrt{g}}\frac{\partial}{\partial r}\left(\frac{\sqrt{g}}{h_r^2} 2 \eta_r^{\mathrm{i}} T^* \frac{\partial u_{||}}{\partial r}\right)\\
&= S_{m u_{||}}^* - u_{||}^* \frac{1}{\sqrt{g}}\frac{\partial}{\partial \theta}\left(\frac{\sqrt{g}}{h_\theta} 2 m n u_\theta\right) - u_{||}^* \frac{1}{\sqrt{g}}\frac{\partial}{\partial r}\left(\frac{\sqrt{g}}{h_r} m n u_r\right)\\
&\qquad - n^* \frac{1}{\sqrt{g}}\frac{\partial}{\partial \theta}\left(\frac{\sqrt{g}}{h_\theta} b_\theta n\right) - p_{\mathrm{n}}^* \frac{1}{\sqrt{g}}\frac{\partial}{\partial \theta}\left(\frac{\sqrt{g}}{h_\theta} b_\theta n_{\mathrm{n,eq}}\right) + T^* \frac{b_\theta}{h_\theta}\frac{\partial p}{\partial \theta}\\
&\qquad - T^* \frac{1}{\sqrt{g}}\frac{\partial}{\partial \theta}\left(\frac{\sqrt{g}}{h_\theta}\frac{5}{2}\left((1 + Z_{\mathrm{i}}) n + n_{\mathrm{n,eq}}\right) b_\theta T\right) + T^* \frac{n_{\mathrm{n,eq}}}{n_{\mathrm{n}}}\frac{b_\theta}{h_\theta}\frac{\partial p_{\mathrm{n}}}{\partial \theta} + \frac{\partial J_1}{\partial u_{||}},
\end{aligned}\tag{B.6}$$

the adjoint neutral pressure diffusion equation,

$$
\begin{aligned}
&-\frac{1}{\sqrt{g}}\frac{\partial}{\partial\theta}\left(\frac{\sqrt{g}}{h_\theta}\left(\frac{\partial n_{\mathrm{n,eq}}}{\partial p_{\mathrm{n}}}u_\theta p_{\mathrm{n}}^* + \left(\frac{5}{2}\frac{\partial n_{\mathrm{n,eq}}}{\partial p_{\mathrm{n}}}u_\theta T - u_{\mathrm{n}\theta}\right)T^*\right)\right)\\
&-\frac{1}{\sqrt{g}}\frac{\partial}{\partial\theta}\left(\frac{\sqrt{g}}{h_\theta^2}\left(D_p^{\mathrm{n}}\frac{\partial p_{\mathrm{n}}^*}{\partial\theta} + \frac{5}{2}D_p^{\mathrm{n}}T\frac{\partial T^*}{\partial\theta} + \frac{D_p^{\mathrm{n}}}{n_{\mathrm{n}}}T^*\frac{\partial p_{\mathrm{n}}}{\partial\theta}\right)\right)\\
&-\frac{1}{\sqrt{g}}\frac{\partial}{\partial r}\left(\frac{\sqrt{g}}{h_r}(-u_{\mathrm{n}r})T^* + \frac{\sqrt{g}}{h_r^2}\left(D_p^{\mathrm{n}}\frac{\partial p_{\mathrm{n}}^*}{\partial r} + \frac{5}{2}D_p^{\mathrm{n}}T\frac{\partial T^*}{\partial r} + \frac{D_p^{\mathrm{n}}}{n_{\mathrm{n}}}T^*\frac{\partial p_{\mathrm{n}}}{\partial r}\right)\right)\\
&= S_{p_{\mathrm{n}}}^* - p_{\mathrm{n}}^*\frac{1}{\sqrt{g}}\frac{\partial}{\partial\theta}\left(\frac{\sqrt{g}}{h_\theta}\frac{\partial n_{\mathrm{n,eq}}}{\partial p_{\mathrm{n}}}u_\theta\right) - T^*\frac{1}{\sqrt{g}}\frac{\partial}{\partial\theta}\left(\frac{\sqrt{g}}{h_\theta}\frac{5}{2}\frac{\partial n_{\mathrm{n,eq}}}{\partial p_{\mathrm{n}}}u_\theta T\right)\\
&\qquad + \frac{D_p^{\mathrm{n}}T^*}{p_{\mathrm{n}}n_{\mathrm{n}}}\left(\frac{1}{h_\theta}\frac{\partial p_{\mathrm{n}}}{\partial\theta}\right)^2 + \frac{D_p^{\mathrm{n}}T^*}{p_{\mathrm{n}}n_{\mathrm{n}}}\left(\frac{1}{h_r}\frac{\partial p_{\mathrm{n}}}{\partial r}\right)^2\\
&\qquad + \left(\frac{\partial n_{\mathrm{n,eq}}}{\partial p_{\mathrm{n}}} - \frac{n_{\mathrm{n,eq}}}{p_{\mathrm{n}}}\right)\frac{u_\theta}{n_{\mathrm{n}}}T^*\frac{1}{h_\theta}\frac{\partial p_{\mathrm{n}}}{\partial\theta} + \frac{\partial J_1}{\partial p_{\mathrm{n}}},
\end{aligned}
\tag{B.7}
$$

and the adjoint energy equation

$$
\begin{aligned}
&-\frac{1}{\sqrt{g}}\frac{\partial}{\partial\theta}\left(\frac{\sqrt{g}}{h_\theta}\frac{5}{2}\sum_a\Gamma_\theta^a T^* + \frac{\sqrt{g}}{h_\theta^2}\kappa_\theta\frac{\partial T^*}{\partial\theta}\right)\\
&-\frac{1}{\sqrt{g}}\frac{\partial}{\partial\theta}\left(\frac{\sqrt{g}}{h_\theta}\left(2b_\theta n u_{||}^* + \frac{\partial n_{\mathrm{n,eq}}}{\partial T}u_\theta p_{\mathrm{n}}^* + \left(\frac{5}{2}\frac{\partial n_{\mathrm{n,eq}}}{\partial T}u_\theta T - 2nu_\theta\right)T^*\right)\right)\\
&-\frac{1}{\sqrt{g}}\frac{\partial}{\partial r}\left(\frac{\sqrt{g}}{h_r}\frac{5}{2}\sum_a\Gamma_r^a T^* + \frac{\sqrt{g}}{h_r^2}\kappa_r\frac{\partial T^*}{\partial r}\right)\\
&= S_E^* - 2b_\theta u_{||}^*\frac{1}{h_\theta}\frac{\partial n}{\partial\theta} - p_{\mathrm{n}}^*\frac{1}{\sqrt{g}}\frac{\partial}{\partial\theta}\left(\frac{\sqrt{g}}{h_\theta}\frac{\partial n_{\mathrm{n,eq}}}{\partial T}u_\theta\right) + 2u_\theta T^*\frac{1}{h_\theta}\frac{\partial n}{\partial\theta}\\
&\quad - T^*\frac{1}{\sqrt{g}}\frac{\partial}{\partial\theta}\left(\frac{\sqrt{g}}{h_\theta}\frac{5}{2}\frac{\partial n_{\mathrm{n,eq}}}{\partial T}u_\theta T\right) - \frac{D_p^{\mathrm{n}}T^*}{p_{\mathrm{n}}}\left(\frac{1}{h_\theta}\frac{\partial p_{\mathrm{n}}}{\partial\theta}\right)^2 - \frac{D_p^{\mathrm{n}}T^*}{p_{\mathrm{n}}}\left(\frac{1}{h_r}\frac{\partial p_{\mathrm{n}}}{\partial r}\right)^2\\
&\quad - T^*\frac{1}{\sqrt{g}}\frac{\partial}{\partial\theta}\left(\frac{\sqrt{g}}{h_\theta}\frac{5}{2}\sum_a\Gamma_\theta^a\right) - T^*\frac{1}{\sqrt{g}}\frac{\partial}{\partial r}\left(\frac{\sqrt{g}}{h_r}\frac{5}{2}\sum_a\Gamma_r^a\right)\\
&\quad + \left(\frac{\partial n_{\mathrm{n,eq}}}{\partial T} + \frac{n_{\mathrm{n,eq}}}{T}\right)\frac{u_\theta}{n_{\mathrm{n}}}T^*\frac{1}{h_\theta}\frac{\partial p_{\mathrm{n}}}{\partial\theta} + \frac{\partial J_1}{\partial T}.
\end{aligned}
\tag{B.8}
$$

The adjoint source terms can be split up in contributions from the linearized source terms S_n and S_z, and contributions from linearizations in transport coefficients. The contributions of S_p are already included in (B.5) – (B.8). The contribution of S_n leads to an adjoint density source

$$\begin{aligned}
S^*_{n_\mathrm{i},n} \quad = \quad & \left(n_\mathrm{n} Z_\mathrm{i} K_\mathrm{i} - 2 n_\mathrm{e} K_\mathrm{r}\right)\left(n^* - p^*_\mathrm{n}\right) + m n_\mathrm{n} (n_\mathrm{e} K_\mathrm{i} + n_\mathrm{i} K_\mathrm{cx}) \frac{\partial u_{\mathrm{n}||}}{\partial n} u^*_{||} \\
& + m \left(n_\mathrm{n} Z_\mathrm{i} K_\mathrm{i} u_{\mathrm{n}||} - 2 n_\mathrm{e} K_\mathrm{r} u_{||} - n_\mathrm{n} K_\mathrm{cx} (u_{||} - u_{\mathrm{n}||})\right) u^*_{||} \\
& - E_\mathrm{i} n_\mathrm{n} Z_\mathrm{i} K_\mathrm{i} T^* + m \left(\left(u_{||} - u_{\mathrm{n}||}\right)^2 + u^2_{\mathrm{n}r} + u^2_{\mathrm{n}\perp}\right) \frac{\partial K}{\partial n} T^* \\
& - 2m \left(\left(u_{||} - u_{\mathrm{n}||}\right) \frac{\partial u_{\mathrm{n}||}}{\partial n} - u_{\mathrm{n}r} \frac{\partial u_{\mathrm{n}r}}{\partial n} - u_{\mathrm{n}\perp} \frac{\partial u_{\mathrm{n}\perp}}{\partial n}\right) K T^*
\end{aligned}$$

with

$$\begin{aligned}
\frac{\partial u_{\mathrm{n}||}}{\partial n} \quad &= \quad \frac{\partial n_{\mathrm{n,eq}}}{\partial n} \frac{u_{||}}{n_\mathrm{n}} - \frac{1}{n_\mathrm{n}} \frac{\partial D^\mathrm{n}_p}{\partial n} \frac{b_\theta}{h_\theta} \frac{\partial p_\mathrm{n}}{\partial \theta}, \\
\frac{\partial u_{\mathrm{n}r}}{\partial n} \quad &= \quad -\frac{1}{n_\mathrm{n}} \frac{\partial D^\mathrm{n}_p}{\partial n} \frac{1}{h_r} \frac{\partial p_\mathrm{n}}{\partial r}, \\
\frac{\partial u_{\mathrm{n}\perp}}{\partial n} \quad &= \quad -\frac{1}{n_\mathrm{n}} \frac{\partial D^\mathrm{n}_p}{\partial n} \frac{b_\phi}{h_\theta} \frac{\partial p_\mathrm{n}}{\partial \theta}, \\
\frac{\partial K}{\partial n} \quad &= \quad \frac{1}{2} n_\mathrm{n} Z_\mathrm{i} K_\mathrm{i} + n_\mathrm{e} K_\mathrm{r} + n_\mathrm{n} K_\mathrm{cx},
\end{aligned}$$

an adjoint momentum source

$$\begin{aligned}
S^*_{m u_{\mathrm{n}||},n} \quad = \quad & -m n_\mathrm{i} (n_\mathrm{e} K_\mathrm{r} + n_\mathrm{n} K_\mathrm{cx}) u^*_{||} + m n_\mathrm{n} (n_\mathrm{e} K_\mathrm{i} + n_\mathrm{i} K_\mathrm{cx}) \frac{\partial u_{\mathrm{n}||}}{\partial u_{||}} u^*_{||} \\
& + 2m \left(u_{||} - u_{\mathrm{n}||}\right) \left(1 - \frac{\partial u_{\mathrm{n}||}}{\partial u_{||}}\right) K T^*,
\end{aligned}$$

with

$$\frac{\partial u_{\mathrm{n}||}}{\partial u_{||}} \quad = \quad \frac{n_{\mathrm{n,eq}}}{n_\mathrm{n}},$$

an adjoint neutral pressure source

$$\begin{aligned}
S^*_{p_\mathrm{n},n} &= n_\mathrm{e}\frac{K_\mathrm{i}}{T}\left(n^* - p^*_\mathrm{n}\right) + m\left(n_\mathrm{e}\frac{K_\mathrm{i}}{T}u_{\mathrm{n}||} - n_\mathrm{i}\frac{K_\mathrm{cx}}{T}(u_{||} - u_{\mathrm{n}||})\right)u^*_{||} \\
&\quad +(mn_\mathrm{e}n_\mathrm{n}K_\mathrm{i} + mn_\mathrm{i}n_\mathrm{n}K_\mathrm{cx})\frac{\partial u_{\mathrm{n}||}}{\partial p_\mathrm{n}}u^*_{||} \\
&\quad -E_\mathrm{i}n_\mathrm{e}\frac{K_\mathrm{i}}{T}T^* + m\left(\left(u_{||} - u_{\mathrm{n}||}\right)^2 + u^2_{\mathrm{n}r} + u^2_{\mathrm{n}\perp}\right)\frac{\partial K}{\partial p_\mathrm{n}}T^* \\
&\quad -2m\left(\left(u_{||} - u_{\mathrm{n}||}\right)\frac{\partial u_{\mathrm{n}||}}{\partial p_\mathrm{n}} - u_{\mathrm{n}r}\frac{\partial u_{\mathrm{n}r}}{\partial p_\mathrm{n}} - u_{\mathrm{n}\perp}\frac{\partial u_{\mathrm{n}\perp}}{\partial p_\mathrm{n}}\right)KT^* \\
&\quad +\frac{1}{\sqrt{g}}\frac{\partial}{\partial\theta}\left(\frac{\sqrt{g}}{h_\theta}(mn_\mathrm{e}n_\mathrm{n}K_\mathrm{i} + mn_\mathrm{i}n_\mathrm{n}K_\mathrm{cx})\frac{D^\mathrm{n}_p}{n_\mathrm{n}}b_\theta u^*_{||}\right) \\
&\quad +\frac{1}{\sqrt{g}}\frac{\partial}{\partial\theta}\left(\frac{\sqrt{g}}{h_\theta}2m\frac{D^\mathrm{n}_p}{n_\mathrm{n}}K\left(u_{\mathrm{n}\perp}b_\phi - \left(u_{||} - u_{\mathrm{n}||}\right)b_\theta\right)T^*\right) \\
&\quad +\frac{1}{\sqrt{g}}\frac{\partial}{\partial r}\left(\frac{\sqrt{g}}{h_r}2m\frac{D^\mathrm{n}_p}{n_\mathrm{n}}Ku_{\mathrm{n}r}T^*\right),
\end{aligned}$$

with

$$\begin{aligned}
\frac{\partial u_{\mathrm{n}||}}{\partial p_\mathrm{n}} &= \left(\frac{\partial n_\mathrm{n,eq}}{\partial p_\mathrm{n}} - \frac{n_\mathrm{n,eq}}{p_\mathrm{n}}\right)\frac{u_{||}}{n_\mathrm{n}} + \frac{1}{n_\mathrm{n}}\frac{D^\mathrm{n}_p}{p_\mathrm{n}}\frac{b_\theta}{h_\theta}\frac{\partial p_\mathrm{n}}{\partial\theta}, \\
\frac{\partial u_{\mathrm{n}r}}{\partial p_\mathrm{n}} &= \frac{1}{n_\mathrm{n}}\frac{D^\mathrm{n}_p}{p_\mathrm{n}}\frac{1}{h_r}\frac{\partial p_\mathrm{n}}{\partial r}, \\
\frac{\partial u_{\mathrm{n}\perp}}{\partial p_\mathrm{n}} &= \frac{1}{n_\mathrm{n}}\frac{D^\mathrm{n}_p}{p_\mathrm{n}}\frac{b_\phi}{h_\theta}\frac{\partial p_\mathrm{n}}{\partial\theta}, \\
\frac{\partial K}{\partial p_\mathrm{n}} &= \frac{1}{2}n_\mathrm{e}\frac{K_\mathrm{i}}{T} + n_\mathrm{i}\frac{K_\mathrm{cx}}{T},
\end{aligned}$$

and an adjoint energy source

$$\begin{aligned}
S^*_{E,n} &= \left(n_\mathrm{e}n_\mathrm{n}\left(\frac{\partial K_\mathrm{i}}{\partial T}-\frac{K_\mathrm{i}}{T}\right)-n_\mathrm{i}n_\mathrm{e}\frac{\partial K_\mathrm{r}}{\partial T}\right)(n^*-p^*_\mathrm{n})\\
&+m\left(n_\mathrm{e}n_\mathrm{n}\left(\frac{\partial K_\mathrm{i}}{\partial T}-\frac{K_\mathrm{i}}{T}\right)u_{\mathrm{n}||}-n_\mathrm{i}n_\mathrm{e}\frac{\partial K_\mathrm{r}}{\partial T}u_{||}\right)u^*_{||}\\
&-mn_\mathrm{n}\left(n_\mathrm{i}\left(\frac{\partial K_\mathrm{cx}}{\partial T}-\frac{K_\mathrm{cx}}{T}\right)(u_{||}-u_{\mathrm{n}||})-(n_\mathrm{e}K_\mathrm{i}+n_\mathrm{i}K_\mathrm{cx})\frac{\partial u_{\mathrm{n}||}}{\partial T}\right)u^*_{||}\\
&-\left(E_\mathrm{i}n_\mathrm{e}n_\mathrm{n}\left(\frac{\partial K_\mathrm{i}}{\partial T}-\frac{K_\mathrm{i}}{T}\right)\right)T^*\\
&+m\left(\left(u_{||}-u_{\mathrm{n}||}\right)^2+u^2_{\mathrm{n}r}+u^2_{\mathrm{n}\perp}\right)\frac{\partial K}{\partial T}T^*\\
&-2m\left(\left(u_{||}-u_{\mathrm{n}||}\right)\frac{\partial u_{\mathrm{n}||}}{\partial T}-u_{\mathrm{n}r}\frac{\partial u_{\mathrm{n}r}}{\partial T}-u_{\mathrm{n}\perp}\frac{\partial u_{\mathrm{n}\perp}}{\partial T}\right)KT^*,
\end{aligned}$$

with

$$\begin{aligned}
\frac{\partial u_{\mathrm{n}||}}{\partial T} &= \left(\frac{\partial n_\mathrm{n,eq}}{\partial T}+\frac{n_\mathrm{n,eq}}{T}\right)\frac{u_{||}}{n_\mathrm{n}}-\frac{1}{n_\mathrm{n}}\left(\frac{\partial D^\mathrm{n}_p}{\partial T}+\frac{D^\mathrm{n}_p}{T}\right)\frac{b_\theta}{h_\theta}\frac{\partial p_\mathrm{n}}{\partial\theta},\\
\frac{\partial u_{\mathrm{n}r}}{\partial T} &= -\frac{1}{n_\mathrm{n}}\left(\frac{\partial D^\mathrm{n}_p}{\partial T}+\frac{D^\mathrm{n}_p}{T}\right)\frac{1}{h_r}\frac{\partial p_\mathrm{n}}{\partial r},\\
\frac{\partial u_{\mathrm{n}\perp}}{\partial T} &= -\frac{1}{n_\mathrm{n}}\left(\frac{\partial D^\mathrm{n}_p}{\partial T}+\frac{D^\mathrm{n}_p}{T}\right)\frac{b_\phi}{h_\theta}\frac{\partial p_\mathrm{n}}{\partial\theta},\\
\frac{\partial K}{\partial T} &= \frac{1}{2}n_\mathrm{e}n_\mathrm{n}\left(\frac{\partial K_\mathrm{i}}{\partial T}-\frac{K_\mathrm{i}}{T}\right)+\frac{1}{2}n_\mathrm{i}n_\mathrm{e}\frac{\partial K_\mathrm{r}}{\partial T}+n_\mathrm{i}n_\mathrm{n}\left(\frac{\partial K_\mathrm{cx}}{\partial T}-\frac{K_\mathrm{cx}}{T}\right).
\end{aligned}$$

The linearized impurity radiation sources lead to the following adjoint terms:

$$\begin{aligned}
S^*_{n_\mathrm{i},z} &= -2c_zn_\mathrm{e}L_zT^*, & S^*_{p_\mathrm{n},z} &= 0,\\
S^*_{mu_{\mathrm{n}||},z} &= 0, & S^*_{E,z} &= -c_zn_\mathrm{i}n_\mathrm{e}\tfrac{\partial L_z}{\partial T}T^*.
\end{aligned}$$

Finally, the effect of variable transport coefficients is translated to the source terms

$$
\begin{aligned}
S^*_{n_\mathrm{i},t} &= -\frac{1}{h_r^2}\frac{\partial \eta_r^\mathrm{i}}{\partial n}\frac{\partial u_{||}^*}{\partial r}\frac{\partial u_{||}}{\partial r} - \frac{1}{h_\theta^2}\frac{\partial D_p^\mathrm{n}}{\partial n}\frac{\partial p_\mathrm{n}^*}{\partial \theta}\frac{\partial p_\mathrm{n}}{\partial \theta} - \frac{1}{h_r^2}\frac{\partial D_p^\mathrm{n}}{\partial n}\frac{\partial p_\mathrm{n}^*}{\partial r}\frac{\partial p_\mathrm{n}}{\partial r} \\
&\quad -\frac{1}{h_\theta^2}\frac{\partial \kappa_\theta}{\partial n}\frac{\partial T^*}{\partial \theta}\frac{\partial T}{\partial \theta} - \frac{1}{h_r^2}\frac{\partial \kappa_r}{\partial n}\frac{\partial T^*}{\partial r}\frac{\partial T}{\partial r} + \frac{\partial \eta_r^\mathrm{i}}{\partial n}\left(\frac{1}{h_r}\frac{\partial u_{||}}{\partial r}\right)^2 T^* \\
&\quad -\frac{5}{2}T\frac{\partial D_p^\mathrm{n}}{\partial n}\left(\frac{1}{h_\theta^2}\frac{\partial T^*}{\partial \theta}\frac{\partial p_\mathrm{n}}{\partial \theta} + \frac{1}{h_r^2}\frac{\partial T^*}{\partial r}\frac{\partial p_\mathrm{n}}{\partial r}\right) \\
&\quad -\frac{T^*}{n_\mathrm{n}}\frac{\partial D_p^\mathrm{n}}{\partial n}\left(\left(\frac{1}{h_\theta}\frac{\partial p_\mathrm{n}}{\partial \theta}\right)^2 + \left(\frac{1}{h_r}\frac{\partial p_\mathrm{n}}{\partial r}\right)^2\right), \\
S^*_{mu_{\mathrm{n}||},t} &= 0, \\
S^*_{p_\mathrm{n},t} &= -\frac{1}{h_\theta^2}\frac{\partial \kappa_\theta}{\partial p_\mathrm{n}}\frac{\partial T^*}{\partial \theta}\frac{\partial T}{\partial \theta} - \frac{1}{h_r^2}\frac{\partial \kappa_r}{\partial p_\mathrm{n}}\frac{\partial T^*}{\partial r}\frac{\partial T}{\partial r}, \\
S^*_{E,t} &= -\frac{1}{h_\theta^2}\frac{\partial \eta_\theta^\mathrm{i}}{\partial T}\frac{\partial u_{||}^*}{\partial \theta}\frac{\partial u_{||}}{\partial \theta} - \frac{1}{h_\theta^2}\frac{\partial D_p^\mathrm{n}}{\partial T}\frac{\partial p_\mathrm{n}^*}{\partial \theta}\frac{\partial p_\mathrm{n}}{\partial \theta} - \frac{1}{h_r^2}\frac{\partial D_p^\mathrm{n}}{\partial T}\frac{\partial p_\mathrm{n}^*}{\partial r}\frac{\partial p_\mathrm{n}}{\partial r} \\
&\quad -\frac{1}{h_\theta^2}\frac{\partial \kappa_\theta}{\partial T}\frac{\partial T^*}{\partial \theta}\frac{\partial T}{\partial \theta} - \frac{1}{h_r^2}\frac{\partial \kappa_r}{\partial T}\frac{\partial T^*}{\partial r}\frac{\partial T}{\partial r} + \frac{\partial \eta_\theta^\mathrm{i}}{\partial T}\left(\frac{1}{h_\theta}\frac{\partial u_{||}}{\partial \theta}\right)^2 T^* \\
&\quad -\frac{5}{2}T\frac{\partial D_p^\mathrm{n}}{\partial T}\left(\frac{1}{h_\theta^2}\frac{\partial T^*}{\partial \theta}\frac{\partial p_\mathrm{n}}{\partial \theta} + \frac{1}{h_r^2}\frac{\partial T^*}{\partial r}\frac{\partial p_\mathrm{n}}{\partial r}\right) \\
&\quad -\frac{T^*}{n_\mathrm{n}}\frac{\partial D_p^\mathrm{n}}{\partial T}\left(\left(\frac{1}{h_\theta}\frac{\partial p_\mathrm{n}}{\partial \theta}\right)^2 + \left(\frac{1}{h_r}\frac{\partial p_\mathrm{n}}{\partial r}\right)^2\right).
\end{aligned}
$$

B.2 Derivation Adjoint Boundary Conditions

The general expression for the adjoint boundary conditions was derived in Chapter 3, Eq. (3.52):

$$\begin{cases} 0 = \nabla_{\mathbf{q}} g + \boldsymbol{\mathcal{C}}_{\mathbf{q}}^T \mathbf{p}^* + \nu_\theta S_{\nabla_\theta \mathbf{q}}^T \mathbf{q}^* + \nu_r S_{\nabla_r \mathbf{q}}^T \mathbf{q}^* \\ \quad -\nu_\theta \left((C_{\mathbf{q}}^\theta)^T \mathbf{q}^* - \mathbf{q}^* \cdot D_{\mathbf{q}}^\theta \frac{1}{h_\theta} \frac{\partial \mathbf{q}}{\partial \theta} + (D^\theta)^T \frac{1}{h_\theta} \frac{\partial \mathbf{q}^*}{\partial \theta} \right) \\ \quad -\nu_r \left((C_{\mathbf{q}}^r)^T \mathbf{q}^* - \mathbf{q}^* \cdot D_{\mathbf{q}}^r \frac{1}{h_r} \frac{\partial \mathbf{q}}{\partial r} + (D^r)^T \frac{1}{h_r} \frac{\partial \mathbf{q}^*}{\partial r} \right), \\ 0 = \boldsymbol{\mathcal{C}}_{\nabla_\theta \mathbf{q}}^T \mathbf{p}^* + \nu_\theta (D^\theta)^T \mathbf{q}^*, \\ 0 = \boldsymbol{\mathcal{C}}_{\nabla_r \mathbf{q}}^T \mathbf{p}^* + \nu_r (D^r)^T \mathbf{q}^*. \end{cases} \tag{B.9}$$

In this section, this general expression is elaborated for the different boundaries in a typical edge plasma simulation. The only terms differing between the different boundaries arise from a possible cost functional defined on that boundary (leading to the contribution $\nabla_{\mathbf{q}} g$) and the specification of the boundary conditions themselves by $\boldsymbol{\mathcal{C}}(\mathbf{q}, \nabla_\theta \mathbf{q}, \nabla_r \mathbf{q})$.

Targets

The boundary conditions at the targets are given by Eq. (3.26):

$$0 = \boldsymbol{\mathcal{C}}_{\mathrm{t}} = \begin{pmatrix} \left(-\mathbf{\Gamma}_{\mathrm{t}} + \mathbf{\Gamma}^{\mathrm{i}}\right) \cdot \boldsymbol{\nu} \\ \pm c_{\mathrm{s}} - u_{||} \\ \left(R\mathbf{\Gamma}_{\mathrm{t}} + \mathbf{\Gamma}^{\mathrm{n}}\right) \cdot \boldsymbol{\nu} \\ \left(-\mathbf{Q}_{\mathrm{t}} + \mathbf{Q}\right) \cdot \boldsymbol{\nu} \end{pmatrix}. \tag{B.10}$$

Linearization with respect to $\mathbf{q}$, $\frac{1}{h_\theta} \frac{\partial \mathbf{q}}{\partial \theta}$ and $\frac{1}{h_r} \frac{\partial \mathbf{q}}{\partial r}$ gives

$$\boldsymbol{\mathcal{C}}_{\mathrm{t},\mathbf{q}}^T = \begin{pmatrix} -u_\theta \nu_\theta + \frac{\partial \mathbf{\Gamma}^{\mathrm{i}}}{\partial n} \cdot \boldsymbol{\nu} & 0 & R u_\theta \nu_\theta + \frac{\partial \mathbf{\Gamma}^{\mathrm{n}}}{\partial n} \cdot \boldsymbol{\nu} & -\delta_{\mathrm{sh}} T u_\theta \nu_\theta + \frac{\partial \mathbf{Q}}{\partial n} \cdot \boldsymbol{\nu} \\ -n b_\theta \nu_\theta + \frac{\partial \mathbf{\Gamma}^{\mathrm{i}}}{\partial u_{||}} \cdot \boldsymbol{\nu} & -1 & R n b_\theta \nu_\theta + \frac{\partial \mathbf{\Gamma}^{\mathrm{n}}}{\partial u_{||}} \cdot \boldsymbol{\nu} & -\delta_{\mathrm{sh}} T n b_\theta \nu_\theta + \frac{\partial \mathbf{Q}}{\partial u_{||}} \cdot \boldsymbol{\nu} \\ 0 & 0 & \frac{\partial \mathbf{\Gamma}^{\mathrm{n}}}{\partial p_{\mathrm{n}}} \cdot \boldsymbol{\nu} & \frac{\partial \mathbf{Q}}{\partial p_{\mathrm{n}}} \cdot \boldsymbol{\nu} \\ 0 & \pm \frac{\partial c_{\mathrm{s}}}{\partial T} & \frac{\partial \mathbf{\Gamma}^{\mathrm{n}}}{\partial T} \cdot \boldsymbol{\nu} & -\delta_{\mathrm{sh}} n u_\theta \nu_\theta + \frac{\partial \mathbf{Q}}{\partial T} \cdot \boldsymbol{\nu} \end{pmatrix},$$

and

$$\begin{aligned}
\boldsymbol{\mathcal{C}}^T_{\mathrm{t},\nabla_\theta \mathbf{q}} &= \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -D^{\mathrm{n}}_p \nu_\theta & -\frac{5}{2} T D^{\mathrm{n}}_p \nu_\theta \\ 0 & 0 & 0 & -\kappa_\theta \nu_\theta \end{pmatrix}, \\
\boldsymbol{\mathcal{C}}^T_{\mathrm{t},\nabla_r \mathbf{q}} &= \begin{pmatrix} -D^{\mathrm{i}} \nu_r & 0 & 0 & -\frac{5}{2}(1+Z_{\mathrm{i}}) T D^{\mathrm{i}} \nu_r \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -D^{\mathrm{n}}_p \nu_r & -\frac{5}{2} T D^{\mathrm{n}}_p \nu_r \\ 0 & 0 & 0 & -\kappa_r \nu_r \end{pmatrix}.
\end{aligned}$$

It is understood that all variables (for example, the (heat) fluxes) are assumed to be functions $\mathbf{q}$,$\nabla_\theta \mathbf{q}$, and $\nabla_r \mathbf{q}$), so partial derivatives with respect to n, $u_{||}$, p_{n} and T are meant for the explicit occurrences of the state variables, and not of their spatial derivatives.

First, consider the last two sets of equations in (B.9). When decoupled, these give eight equations in only four unknowns. However, three equations appear twice and one equation is trivial. The remaining equations lead to the conditions

$$\left\{ \begin{array}{rcl} n^*_S &=& n^*, \\ u^*_{||} &=& 0, \\ p^*_{\mathrm{n},S} &=& p^*_{\mathrm{n}}, \\ T^*_S &=& T^*. \end{array} \right. \tag{B.11}$$

In (B.11), the first, third and fourth rows define the adjoint multipliers on the (target) boundary, while the second row is the actual boundary condition for the adjoint parallel velocity.

Next, the first equation in (B.9) is considered. By taking $u^*_{||} = 0$ some terms immediately cancel. After further elaboration, the following conditions are retained:

$$\begin{cases} 0 = \frac{\partial g}{\partial n} - u_\theta n^* \nu_\theta + R u_\theta p_\mathrm{n}^* \nu_\theta - \delta_\mathrm{sh} u_\theta T T^* \nu_\theta - m D^\mathrm{i} u_{||} \frac{1}{h_r} \frac{\partial u_{||}^*}{\partial r} \nu_r \\ \quad -\frac{5}{2}(1+Z_\mathrm{i}) D^\mathrm{i} T \frac{1}{h_r} \frac{\partial T^*}{\partial r} \nu_r + 2 u_\theta T T^* \nu_\theta - D^\mathrm{i} \frac{1}{h_r} \frac{\partial n^*}{\partial r} \nu_r, \\[2ex] u_{||,S}^* = \frac{\partial g}{\partial u_{||}} - n b_\theta n^* \nu_\theta + R n b_\theta p_\mathrm{n}^* \nu_\theta - \delta_\mathrm{sh} n b_\theta T T^* \nu_\theta \\ \quad +\eta_\theta^\mathrm{i} \left(2T^* \frac{1}{h_\theta} \frac{\partial u_{||}}{\partial \theta} - \frac{1}{h_\theta} \frac{\partial u_{||}^*}{\partial \theta} \right) \nu_\theta + \eta_r^\mathrm{i} \left(2T^* \frac{1}{h_r} \frac{\partial u_{||}}{\partial r} - \frac{1}{h_r} \frac{\partial u_{||}^*}{\partial r} \right) \nu_r, \\[2ex] 0 = \frac{\partial g}{\partial p_\mathrm{n}} - 2KmT^* \frac{D_p^\mathrm{n}}{n_\mathrm{n}} \left((u_{\mathrm{n}\theta} - u_\theta)\nu_\theta + u_{\mathrm{n}r} \nu_r \right) - D_p^\mathrm{n} \left(\frac{1}{h_\theta} \frac{\partial p_\mathrm{n}^*}{\partial \theta} \nu_\theta + \frac{1}{h_r} \frac{\partial p_\mathrm{n}^*}{\partial r} \nu_r \right) \\ \quad +2 \left(u_{\mathrm{n}\theta} \nu_\theta + u_{\mathrm{n}r} \nu_r \right) T^* - \frac{n_\mathrm{n,eq}}{n_\mathrm{n}} u_\theta T^* \nu_\theta - \frac{5}{2} D_p^\mathrm{n} T \left(\frac{1}{h_\theta} \frac{\partial T^*}{\partial \theta} \nu_\theta + \frac{1}{h_r} \frac{\partial T^*}{\partial r} \nu_r \right), \\[2ex] 0 = \frac{\partial g}{\partial T} - \delta_\mathrm{sh} n u_\theta T^* \nu_\theta \pm \frac{\partial c_\mathrm{s}}{\partial T} u_{||,S}^* + 2 n u_\theta T^* \nu_\theta - \kappa_\theta \frac{1}{h_\theta} \frac{\partial T^*}{\partial \theta} \nu_\theta - \kappa_r \frac{1}{h_r} \frac{\partial T^*}{\partial r} \nu_r. \end{cases} \tag{B.12}$$

The second equation defines the adjoint multiplier $u_{||,S}^*$, while the other three define the adjoint ion, neutral and energy flux, respectively.

It is seen that the derivatives of a cost functional on the boundary appear as forcing terms in the different adjoint boundary conditions. In this thesis, focus is on optimizing the spreading of the heat load at the targets. The specific form of the cost functional is discussed in the different chapters.

Core

The core boundary conditions are, from (3.27),

$$0 = \boldsymbol{\mathcal{C}}_\mathrm{c} = \begin{pmatrix} n_\mathrm{c} - n \\ -u_{||} \\ -\Gamma_r^\mathrm{n} \\ Q_{r,\mathrm{c}} - Q_r \end{pmatrix}. \tag{B.13}$$

After linearization, the matrices

$$\boldsymbol{\mathcal{C}}_{\mathrm{c},\mathbf{q}}^T = \begin{pmatrix} -1 & 0 & 0 & \frac{\partial \kappa_r}{\partial n} \frac{1}{h_r} \frac{\partial T}{\partial r} \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & \frac{\partial \kappa_r}{\partial p_\mathrm{n}} \frac{1}{h_r} \frac{\partial T}{\partial r} \\ 0 & 0 & 0 & -\frac{5}{2} \sum_a \Gamma_r^a + \frac{\partial \kappa_r}{\partial T} \frac{1}{h_r} \frac{\partial T}{\partial r} \end{pmatrix},$$

and

$$\boldsymbol{\mathcal{C}}_{\mathrm{c},\nabla_r \mathbf{q}}^T = \begin{pmatrix} 0 & 0 & 0 & \frac{5}{2}(1+Z_\mathrm{i}) T D^\mathrm{i} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & D_p^\mathrm{n} & \frac{5}{2} T D_p^\mathrm{n} \\ 0 & 0 & 0 & \kappa_r \end{pmatrix}.$$

are found.

Since $\nu_\theta = 0$ at the core, only the first and the last set of equations in (B.9) have to be used, giving eight equations for eight unknowns. The last equation leads to

$$\left\{ \begin{array}{rcl} n^* & = & 0, \\ u^*_{||} & = & 0, \\ p^*_{\mathrm{n},S} & = & p^*_{\mathrm{n}}, \\ T^*_S & = & T^*. \end{array} \right. \tag{B.14}$$

Combining this with the first equation in (B.9) then gives

$$\left\{ \begin{array}{rcl} n^*_S & = & \frac{\partial g}{\partial n} + D^{\mathrm{i}} \frac{1}{h_r} \frac{\partial n^*}{\partial r} + \frac{5}{2}(1 + Z_{\mathrm{i}}) T D^{\mathrm{i}} \frac{1}{h_r} \frac{\partial T^*}{\partial r}, \\ u^*_{||,S} & = & \frac{\partial g}{\partial u_{||}} + \eta^{\mathrm{i}}_r \frac{1}{h_r} \frac{\partial u^*_{||}}{\partial r} - 2\eta^{\mathrm{i}}_r T^* \frac{1}{h_r} \frac{\partial u_{||}}{\partial r}, \\ 0 & = & \frac{\partial g}{\partial p_{\mathrm{n}}} + D^{\mathrm{n}}_p \frac{1}{h_r} \frac{\partial p^*_{\mathrm{n}}}{\partial r} + \frac{5}{2} T D^{\mathrm{n}}_p \frac{1}{h_r} \frac{\partial T^*}{\partial r}, \\ 0 & = & \frac{\partial g}{\partial T} + \kappa_r \frac{1}{h_r} \frac{\partial T^*}{\partial r}. \end{array} \right. \tag{B.15}$$

In some cases, there can be neutral leakage to the core. Then, the neutral boundary condition is modified to $-\alpha_{\mathrm{p}} c_{\mathrm{n}} n_{\mathrm{n}} - \Gamma^{\mathrm{n}}_r = 0$ [21]. The exercise above can be repeated to find the modified adjoint boundary conditions for neutral density and energy

$$\left\{ \begin{array}{rcl} 0 & = & \frac{\partial g}{\partial p_{\mathrm{n}}} - \alpha_{\mathrm{p}} \frac{c_{\mathrm{n}}}{T} p^*_{\mathrm{n},S} - 2u_{\mathrm{n}r} T^* \left(1 - mK \frac{D^{\mathrm{n}}_p}{n_{\mathrm{n}}}\right) + D^{\mathrm{n}}_p \frac{1}{h_r} \frac{\partial p^*_{\mathrm{n}}}{\partial r} + \frac{5}{2} T D^{\mathrm{n}}_p \frac{1}{h_r} \frac{\partial T^*}{\partial r}, \\ 0 & = & \frac{\partial g}{\partial T} - \alpha_{\mathrm{p}} \left(\frac{\partial c_{\mathrm{n}}}{\partial T} - \frac{c_{\mathrm{n}}}{T}\right) n_{\mathrm{n}} p^*_{\mathrm{n},S} + \kappa_r \frac{1}{h_r} \frac{\partial T^*}{\partial r}. \end{array} \right. \tag{B.16}$$

The terms related to a possible cost functional are kept in the equations for generality. Although not considered in this thesis, several useful optimal design problems may involve such a cost functional, for example divertor design for maximum core power subject to target heat load constraints. It is recalled that in general, the adjoint problem provides sensitivities of output quantities with respect to all input quantities. Therefore, adjoints may also be used directly in for example optimal control problems where the aim is to control gas puff and pumping speed in order to achieve a certain core (separatrix) density.

Wall

At the outer wall, decay length conditions are combined with recycling for the neutrals,

$$0 = \boldsymbol{\mathcal{C}}_{\mathrm{w}} = \begin{pmatrix} -D^{\mathrm{i}} \frac{n}{\lambda_n} - D^{\mathrm{i}} \nabla_r n \\ -\eta^{\mathrm{i}}_r \nabla_r u_{||} \\ R\Gamma^{\mathrm{i}}_r + \Gamma^{\mathrm{n}}_r \\ -\kappa_r \frac{T}{\lambda_T} - \kappa_r \nabla_r T \end{pmatrix}. \tag{B.17}$$

Linearization with respect to the plasma state variables gives

$$\boldsymbol{\mathcal{C}}^{T}_{\mathrm{w},\mathbf{q}} = \begin{pmatrix} -\frac{D^{\mathrm{i}}}{\lambda_n} & 0 & -\frac{\partial D^{\mathrm{n}}_p}{\partial n}\frac{1}{h_r}\frac{\partial p_{\mathrm{n}}}{\partial r} & -\frac{\partial \kappa_r}{\partial n}\left(\frac{T}{\lambda_T}+\frac{1}{h_r}\frac{\partial T}{\partial r}\right) \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{\partial \kappa_r}{\partial p_{\mathrm{n}}}\left(\frac{T}{\lambda_T}+\frac{1}{h_r}\frac{\partial T}{\partial r}\right) \\ 0 & 0 & -\frac{\partial D^{\mathrm{n}}_p}{\partial T}\frac{1}{h_r}\frac{\partial p_{\mathrm{n}}}{\partial r} & -\frac{\partial \kappa_r}{\partial T}\left(\frac{T}{\lambda_T}+\frac{1}{h_r}\frac{\partial T}{\partial r}\right) - \frac{\kappa_r}{\lambda_n} \end{pmatrix},$$

and

$$\boldsymbol{\mathcal{C}}^{T}_{\mathrm{w},\nabla_r\mathbf{q}} = \begin{pmatrix} -D^{\mathrm{i}} & 0 & -RD^{\mathrm{i}} & 0 \\ 0 & -\eta^{\mathrm{i}}_r & 0 & 0 \\ 0 & 0 & -D^{\mathrm{n}}_p & 0 \\ 0 & 0 & 0 & -\kappa_r \end{pmatrix}.$$

The outer wall is a boundary with $\nu_\theta = 0$ and $\nu_r = 1$. Therefore, the first and third equations in (B.9) have to be elaborated. The third equation leads to

$$\begin{cases} n^*_S &= n^* + mu_{||}u^*_{||} + \frac{5}{2}(1+Z_{\mathrm{i}})TT^* - Rp^*_{\mathrm{n},S}, \\ u^*_{||,S} &= u^*_{||}, \\ p^*_{\mathrm{n},S} &= p^*_{\mathrm{n}} + \frac{5}{2}TT^*, \\ T^*_S &= T^*. \end{cases} \tag{B.18}$$

while the first equation then gives

$$\begin{cases} 0 = \frac{\partial g}{\partial n} - \frac{\partial \kappa_r}{\partial n}\frac{TT^*_S}{\lambda_T} - D^{\mathrm{i}}\left(mu_{||}\frac{1}{h_r}\frac{\partial u^*_{||}}{\partial r} + \frac{5}{2}(1+Z_{\mathrm{i}})T\frac{1}{h_r}\frac{\partial T^*}{\partial r} + \frac{n^*_S}{\lambda_n}\right) - D^{\mathrm{i}}\frac{1}{h_r}\frac{\partial n^*}{\partial r}, \\ 0 = \frac{\partial g}{\partial u_{||}} - mnu_r u^*_{||} - \eta^{\mathrm{i}}_r\frac{1}{h_r}\frac{\partial u^*_{||}}{\partial r}, \\ 0 = \frac{\partial g}{\partial p_{\mathrm{n}}} - \frac{\partial \kappa_r}{\partial p_{\mathrm{n}}}\frac{TT^*_S}{\lambda_T} + 2u_{\mathrm{n}r}T^*\left(1 - mK\frac{D^{\mathrm{n}}_p}{n_{\mathrm{n}}}\right) - \frac{5}{2}D^{\mathrm{n}}_p T\frac{1}{h_r}\frac{\partial T^*}{\partial r} - D^{\mathrm{n}}_p\frac{1}{h_r}\frac{\partial p^*_{\mathrm{n}}}{\partial r}, \\ 0 = \frac{\partial g}{\partial T} - \kappa_r\left(\frac{T}{\kappa_r}\frac{\partial \kappa_r}{\partial T} + 1\right)\frac{T^*_S}{\lambda_T} - \frac{5}{2}\sum_a \Gamma^a_r T^* - \kappa_r\frac{1}{h_r}\frac{\partial T^*}{\partial r}. \end{cases} \tag{B.19}$$

Again, terms related to an objective functional are kept in the expressions. Interesting measurements at the outer wall include sputtering yield and particle and energy fluxes to plasma-facing components and diagnostic mirrors.

Private Flux Boundary

Lastly, the private flux boundary is treated. It is similar to the wall boundary described above. Some signs have switched because now $\nu_r = -1$, and additionally there is neutral pumping:

$$0 = \boldsymbol{\mathcal{C}}_{\mathrm{p}} = \begin{pmatrix} -D^{\mathrm{i}}\frac{n}{\lambda_n} + D^{\mathrm{i}}\nabla_r n \\ \eta_r^{\mathrm{i}}\nabla_r u_{||} \\ -\alpha_{\mathrm{p}} c_{\mathrm{n}} n_{\mathrm{n}} - R\Gamma_r^{\mathrm{i}} - \Gamma_r^{\mathrm{n}} \\ -\kappa_r\frac{T}{\lambda_T} + \kappa_r\nabla_r T \end{pmatrix}. \tag{B.20}$$

Linearization with respect to the plasma state variables gives

$$\boldsymbol{\mathcal{C}}_{\mathrm{p},\mathbf{q}}^{T} = \begin{pmatrix} -\frac{D^{\mathrm{i}}}{\lambda_n} & 0 & \frac{\partial D_p^{\mathrm{n}}}{\partial n}\frac{1}{h_r}\frac{\partial p_{\mathrm{n}}}{\partial r} & \frac{\partial \kappa_r}{\partial n}\left(\frac{1}{h_r}\frac{\partial T}{\partial r} - \frac{T}{\lambda_T}\right) \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -\alpha_{\mathrm{p}}\frac{c_{\mathrm{n}}}{T} & \frac{\partial \kappa_r}{\partial p_{\mathrm{n}}}\left(\frac{1}{h_r}\frac{\partial T}{\partial r} - \frac{T}{\lambda_T}\right) \\ 0 & 0 & \frac{\partial D_p^{\mathrm{n}}}{\partial T}\frac{1}{h_r}\frac{\partial p_{\mathrm{n}}}{\partial r} - \alpha_{\mathrm{p}}\left(\frac{\partial c_{\mathrm{n}}}{\partial T} - \frac{c_{\mathrm{n}}}{T}\right) n_{\mathrm{n}} & \frac{\partial \kappa_r}{\partial T}\left(\frac{1}{h_r}\frac{\partial T}{\partial r} - \frac{T}{\lambda_T}\right) - \frac{\kappa_r}{\lambda_n} \end{pmatrix},$$

and

$$\boldsymbol{\mathcal{C}}_{\mathrm{p},\nabla_r\mathbf{q}}^{T} \quad = \quad \begin{pmatrix} D^{\mathrm{i}} & 0 & RD^{\mathrm{i}} & 0 \\ 0 & \eta_r^{\mathrm{i}} & 0 & 0 \\ 0 & 0 & D_p^{\mathrm{n}} & 0 \\ 0 & 0 & 0 & \kappa_r \end{pmatrix}.$$

The third equation in (B.9) gives

$$\begin{cases} n_S^* &=& n^* + m u_{||} u_{||}^* + \frac{5}{2}(1+Z_{\mathrm{i}})TT^* - Rp_{\mathrm{n},S}^*, \\ u_{||,S}^* &=& u_{||}^*, \\ p_{\mathrm{n},S}^* &=& p_{\mathrm{n}}^* + \frac{5}{2}TT^*, \\ T_S^* &=& T^*, \end{cases} \tag{B.21}$$

and the first equation then gives

$$\begin{cases} 0 &=& \frac{\partial g}{\partial n} - \frac{\partial \kappa_r}{\partial n}\frac{TT_S^*}{\lambda_T} + D^{\mathrm{i}}\left(m u_{||}\frac{1}{h_r}\frac{\partial u_{||}^*}{\partial r} + \frac{5}{2}(1+Z_{\mathrm{i}})T\frac{1}{h_r}\frac{\partial T^*}{\partial r} - \frac{n_S^*}{\lambda_n}\right) + D^{\mathrm{i}}\frac{1}{h_r}\frac{\partial n^*}{\partial r}, \\ 0 &=& \frac{\partial g}{\partial u_{||}} + m n u_r u_{||}^* + \eta_r^{\mathrm{i}}\frac{1}{h_r}\frac{\partial u_{||}^*}{\partial r}, \\ 0 &=& \frac{\partial g}{\partial p_{\mathrm{n}}} - \alpha_{\mathrm{p}}\frac{c_{\mathrm{n}}}{T}p_{\mathrm{n},S}^* - \frac{\partial \kappa_r}{\partial p_{\mathrm{n}}}\frac{TT_S^*}{\lambda_T} - 2u_{\mathrm{n}r}T^*\left(1 - mK\frac{D_p^{\mathrm{n}}}{n_{\mathrm{n}}}\right) \\ && \quad + \frac{5}{2}D_p^{\mathrm{n}}T\frac{1}{h_r}\frac{\partial T^*}{\partial r} + D_p^{\mathrm{n}}\frac{1}{h_r}\frac{\partial p_{\mathrm{n}}^*}{\partial r}, \\ 0 &=& \frac{\partial g}{\partial T} - \alpha_{\mathrm{p}}\left(\frac{\partial c_{\mathrm{n}}}{\partial T} - \frac{c_{\mathrm{n}}}{T}\right) n_{\mathrm{n}} p_{\mathrm{n},S}^* - \kappa_r\left(\frac{T}{\kappa_r}\frac{\partial \kappa_r}{\partial T} + 1\right)\frac{T_S^*}{\lambda_T} + \frac{5}{2}\sum_a \Gamma_r^a T^* + \kappa_r\frac{1}{h_r}\frac{\partial T^*}{\partial r}. \end{cases} \tag{B.22}$$

Cost functionals defined at this pumping boundary can be related to a.o. optimal pumping.

Bibliography

[1] Anderson, W. K., and Venkatakrishnan, V. Aerodynamic design optimization on unstructured grids with a continuous adjoint formulation. *Computers & Fluids 28* (1999), 443–480.

[2] Arian, E., and Ta'asan, S. Shape optimization in one shot. In *Proceedings of a Workshop on Optimal Design and Control* (Blacksburg, April 1994).

[3] Arian, E., and Ta'asan, S. Analysis of the hessian for aerodynamic optimization: Inviscid flow. Tech. rep., Institute for Computer Applications in Science and Engineering (ICASE), Report No. 96-28, 1996.

[4] Aumeunier, M.-H., Travère, J. M., and Reichle, R. A realistic photonic modelling for the evaluation of the infrared reflections in the metallic environment of ITER. 20th ITPA Topical Group Meeting on Diagnostics, May 2011.

[5] Baelmans, M. *Code Improvements and Applications of a Two-dimensional Edge Plasma Model for Toroidal Devices.* PhD thesis, Departement Mechanica, Faculteit der Toegepaste Wetenschappen, Katholieke Universiteit Leuven, October 1993.

[6] Baelmans, M., Börner, P., Dekeyser, W., and Reiter, D. Tokamak plasma edge modelling including the main chamber wall. *Nuclear Fusion 51* (2011), 083023.

[7] Baelmans, M., Reiter, D., and Weynants, R. R. New developments in plasma edge modelling with particular emphasis on drift flows and electric fields. *Contributions to Plasma Physics 36* (1996), 117–126.

[8] Baelmans, M., Reiter, D., Weynants, R. R., and Schneider, R. Computational assessment of effects of electric fields and currents in

tokamak edge plasmas. *Journal of Nuclear Materials 220–222* (1995), 982–986.

[9] Balescu, R. *Transport Processes in Plasmas*, vol. 1. North-Holland, Amsterdam, 1988.

[10] Bischof, C., Corliss, A. C. G., Griewank, A., and Hovland, P. Adifor: Generating derivative codes from fortran programs. *Scientific Programming 1*, 1 (1992), 11–29.

[11] Blommaert, M. Optimization of the magnetic divertor configuration for nuclear fusion reactors. Master's thesis, Department of Mechanical Engineering, KU Leuven, 2012.

[12] Borzì, A., and Schulz, V. Multigrid methods for PDE optimization. *SIAM Review 51* (2009), 361–395.

[13] Borzì, A., and Schulz, V. *Computational Optimization of Systems Governed by Partial Differential Equations.* Computational Science & Egineering. SIAM, Philadelphia, 2012.

[14] Braams, B. J. Numerical studies of the two-dimensional scrape-off layer. In *Proc. 11th Eur. Conf. on Contr. Fus. and Plasma Phys.* (Aachen, 1983), p. 431.

[15] Braams, B. J. *Computational Studies in Tokamak Equilibrium and Transport.* PhD thesis, Rijksuniversiteit Utrecht, 1986.

[16] Braginskii, S. I. Transport processes in a plasma. *Reviews of Plasma Physics 1* (1965), 205–311.

[17] Brezillon, J., and Gauger, N. 2D and 3D aerodynamic shape optimisation using the adjoint approach. *Aerospace Science and Technology 8* (2004), 715–727.

[18] Bucalossi, J., Loarer, T., Philipps, V., Brennan, D., Coad, P., Grisolia, C., Stagg, R., and contributors to the EFDA-JET work programme. Particle balance studies in JET. *Proceedings of the 28th EPS Conference on Controlled Fusion and Plasma Physics, ECA 25A* (2001), 1629–1632.

[19] Choi, K. K., and Kim, N. H. *Structural Sensitivity Analysis and Optimization 1: Linear Systems.* Mechanical Engineering Series. Springer, New York, 2005.

[20] Coster, D. SOLPS5.0, February 2013.

[21] Coster, D. P., Bonnin, X., Braams, B., Reiter, D., Schneider, R., and the ASDEX Upgrade Team. Simulation of the edge plasma in tokamaks. *Physica Scripta 2004*, T108 (2004), 7–13.

[22] Coster, D. P., Kim, J. W., Haas, G., Kurzan, B., Murmann, H., Neuhauser, J., Salzmann, H., Schneider, R., Schneider, W., Schweinzer, J., and Team, A. U. Automatic evaluation of edge transport coefficients with B2-SOLPS5.0. *Contributions to Plasma Physics 40* (2000), 334–339.

[23] De Schutter, J. Sensitivity analysis of optimal divertor configurations in nuclear fusion reactors. Master's thesis, Departement Mechanica, Faculteit Ingenieurswetenschappen, KU Leuven, 2013.

[24] Dekeyser, W., Baelmans, M., Reiter, D., Börner, P., and Kotov, V. 2D edge plasma modeling extended up to the main chamber. *Journal of Nuclear Materials 415* (2011), S584–S588.

[25] Dekeyser, W., Baelmans, M., Voskoboynikov, S., Rozhansky, V., Reiter, D., Wiesen, S., Kotov, V., and Boerner, P. B2-B2.5 code benchmarking, part I: Modeling equations, code "dis-integration". Tech. rep., Report Forschungszentrum Juelich, JUEL-4337, ISSN 0944-2952, 2011.

[26] Dekeyser, W., Reiter, D., and Baelmans, M. Optimal shape design for divertors. *International Journal of Computational Science and Engineering*. In press.

[27] Dekeyser, W., Reiter, D., and Baelmans, M. Adapting computational optimization concepts from aeronautics to nuclear fusion reactor design. *EPJ Web of Conferences 33* (2012), 03009.

[28] Dekeyser, W., Reiter, D., and Baelmans, M. Divertor design through shape optimization. *Contributions to Plasma Physics 52*, 5-6 (2012), 544–549.

[29] Delfour, M. C., and Zolésio, J.-P. *Shapes and Geometries: Metrics, Analysis, Differential Calculus, and Optimization*, 2 ed. Advances in Design and Control. SIAM, Philadelphia, 2011.

[30] Delport, S. *Optimal Control of a Turbulent Mixing Layer*. PhD thesis, Departement of Mechanical Engineering, KU Leuven, 2010.

[31] Désidéri, J.-A. *Numerical methods for scientific computing, Variational problems and applications*. CIMNE, Barcelona, 2003, ch. Hierarchical optimum-shape algorithms using embedded Bézier parameterizations.

[32] Désidéri, J.-A., El Majd, B. A., and Janka, A. Nested and self-adaptive Bézier parameterizations for shape optimization. *Journal of Computational Physics 224* (2007), 117–131.

[33] D'haeseleer, W. D., Hitchon, W. N. G., Callen, J. D., and Shohet, J. L. *Flux Coordinates and Magnetic Field Structure.* Springer-Verlag, Berlin, 1991.

[34] Ferziger, J. H., and Peric, M. *Computational Methods for Fluid Dynamics.* Springer-Verlag, Berlin, 2002.

[35] Frerichs, H. 3D numerische Parameterstudien zum Verunreinigungstransport in TEXTOR-DED. Master's thesis, Institut für Theoretische Physik A, RWTH Aachen, 2006.

[36] Fundamenski, W. *Power Exhaust in Fusion Plasmas.* Cambridge University Press, Cambridge, 2010.

[37] Furubayashi, M., Hoshino, K., Toma, M., Hatayama, A., Coster, D., Schneider, R., Bonnin, X., Kawashima, H., Asakura, N., and Suzuki, Y. Comparison of kinetic and fluid neutral models for attached and detached state. *Journal of Nuclear Materials 390–391*, 0 (2009), 295–298. Proceedings of the 18th International Conference on Plasma-Surface Interactions in Controlled Fusion Devices.

[38] Gao, Z. M., Ma, Y. C., and Zhuang, H. W. Shape optimization for Navier-Stokes flow. *Inverse Problems in Science and Engineering 16*, 5 (July 2008), 583–616.

[39] Gaspar, J., Rigollet, F., Gardarein, J.-L., Niliot, C. L., and Corre, Y. Characterization of time varying thermophysical property of a surface layer: Numerical feasibility for JET tokamak tiles. *International Journal of Heat and Mass Transfer 56* (2013), 147–157.

[40] Gauger, N., Walther, A., Moldenhauer, C., and Widhalm, M. Automatic differentiation of an entire design chain for aerodynamic shape optimization. *Notes on Numerical Fluid Mechanics and Multidisciplinary Design 96* (2007), 454–461.

[41] Gerhauser, H., Claassen, H. A., and Reiter, D. Modelling the tokamak edge plasma with a coupled 2-D fluid/monte carlo code. *Contributions to Plasma Physics 28* (1988), 359–364.

[42] Ghoos, K. The effect of noise on the convergence of the coupled B2-Eirene code system. Master's thesis, Department of Mechanical Engineering, KU Leuven, 2013.

[43] Giannakoglou, K. C., and Papadimitriou, D. I. *Optimization and Computational Fluid Dynamics.* Springer, Berlin, 2008, ch. Adjoint Methods for Shape Optimization, pp. 79–108.

[44] Giannakoglou, K. C., Papadimitriou, D. I., and Kampolis, I. C. Aerodynamic shape design using evolutionary algorithms and new gradient-assisted metamodels. *Computer Methods in Applied Mechanics and Engineering 195* (2006), 6312–6329.

[45] Giles, M. On the iterative solution of adjoint equations, 2000.

[46] Giles, M. B., and Pierce, N. A. Adjoint equations in CFD: duality, boundary conditions and solution behaviour. In *13th Computational Fluid Dynamics Conference* (1997).

[47] Giles, M. B., and Pierce, N. A. An introduction to the adjoint approach to design. *Flow, Turbulence and Combustion 65* (2000), 393–415.

[48] Goldston, R. J., and Rutherford, P. H. *Introduction to Plasma Physics.* Taylor & Francis, 1995.

[49] Greenwood, J. The correct and incorrect generation of a cosine distribution of scattered particles for Monte-Carlo modelling of vacuum systems. *Vacuum 67* (2002), 217–222.

[50] Griewank, A., Juedes, D., and Utke, J. ADOL-C: A package for the automatic differentiation of algorithms written in C/C++. *ACM Transactions on Mathematical Software 22* (1996), 131–167.

[51] Guillemaut, C., Pitts, R. A., Kukushkin, A. S., and O'Mullane, M. Radiative power loading in the ITER divertor. *Fusion Engineering and Design 86* (2011), 2954–2964.

[52] Haslinger, J., and Mäkinen, R. A. E. *Introduction to Shape Optimization: Theory, Approximation, and Computation.* Advances in Design and Control. SIAM, Philadelphia, 2003.

[53] Hazra, S. B. Multigrid one-shot method for aerodynamic shape optimization. *SIAM Journal on Scientific Computing 30*, 3 (2008), 1527–1547.

[54] Hazra, S. B. Multigrid one-shot method for state constrained aerodynamic shape optimization. *SIAM Journal on Scientific Computing 30*, 6 (2008), 3220–3248.

[55] Hazra, S. B., and Schulz, V. Simultaneous pseudo-timestepping for PDE-model based optimization problems. *BIT Numerical Mathematics 44* (2004), 457–472.

[56] Hazra, S. B., and Schulz, V. Simultaneous pseudo-timestepping for aerodynamic shape optimization problems with state constraints. *SIAM Journal on Scientific Computing 28*, 3 (2006), 1078–1099.

[57] Hazra, S. B., Schulz, V., Brezillon, J., and Gauger, N. R. Aerodynamic shape optimization using simultaneous pseudo-timestepping. *Journal of Computational Physics 204* (2005), 16–64.

[58] Hicks, R. M., and Henne, P. A. Wing design by numerical optimization. *Journal of Aircraft 15* (1978), 407–412.

[59] Hinze, M., Pinnau, R., Ulbrich, M., and Ulbrich, S. *Optimization with PDE constraints*. Mathematical Modelling: Theory and Applications. Springer, Berlin, 2009.

[60] Hoshino, K., Toma, M., Hatayama, A., Coster, D. P., Bonnin, X., Schneider, R., Kawashima, H., Asakura, N., and Suzuki, Y. Benchmarking kinetic and fluid neutral models with drift effects. *Contributions to Plasma Physics 48*, 1-3 (2008), 136–140.

[61] Ito, K., Kunisch, K., and Peichl, G. H. Variational approach to shape derivatives. *ESAIM: Control, Optimisation and Calculus of Variations 14* (2008), 517–539.

[62] Jameson, A. Aerodynamic design via control theory. *Journal of Scientific Computing 3*, 3 (1988), 233–260.

[63] Jameson, A. Optimum aerodynamic design using CFD and control theory. In *AIAA Paper 95-1729* (1995).

[64] Jameson, A., and Kim, S. Reduction of the adjoint gradient formula in the continuous limit. In *41 st AIAA Aerospace Sciences Meeting, AIAA Paper 2003-0040* (2003).

[65] Jameson, A., Martinelli, L., and Pierce, N. A. Optimum aerodynamic design using the Navier-Stokes equations. *Theoretical and Computational Fluid Dynamics 10* (1998), 213–237.

[66] Kanschat, G., Meinköhn, E., Rannacher, R., and Wehrse, R. *Numerical Methods in Multidimensional Radiative Transfer*. Springer-Verlag, 2009, ch. Introduction: The Radiation Field and its Transfer Equation, pp. 1–18.

[67] Karakasis, M. K., and Giannakoglou, K. C. On the use of metamodel-assisted multi-objective evolutionary algorithms. *Engineering Optimization 38* (2006), 941–957.

[68] Karakasis, M. K., Giotis, A. P., and Giannakoglou, K. C. Inexact information aided low-cost distributed genetic algorithms for aerodynamic shape optimization. *International Journal for Numerical Methods in Fluids 43* (2003), 1149–1166.

[69] Kasumba, H., and Kunisch, K. On shape sensitivity analysis of the cost functional without shape sensitivity of the state variable. *Control & Cybernetics 40* (2011), 989.

[70] Knoll, D. Computational study of ITER-like dissipative divertor plasmas in the collisional limit. *Nuclear Fusion 38*, 1 (1998), 133–146.

[71] Knoll, D., and McHugh, P. NEWEDGE: a 2D fully implicit edge plasma fluid code for advanced physics and complex geometries. *Journal of Nuclear Materials 196–198*, 0 (1992), 352–356. Proceedings of the Tenth International Conference on Plasma-Surface Interactions in Controlled Fusion Devices.

[72] Knoll, D. A., Krasheninnikov, S. I., McHugh, P. R., and Sigmar, D. J. Characterization of a partially-ionized, detached, divertor plasma. *Physics of Plasmas 3*, 9 (1996), 3358–3368.

[73] Knoll, D. A., McHugh, P. R., Krasheninnikov, S. I., and Sigmar, D. J. Simulation of dense recombining divertor plasmas with a Navier–Stokes neutral transport model. *Physics of Plasmas 3*, 1 (1996), 293–303.

[74] Kotschenreuther, M., Valanju, P. M., Mahajan, S. M., and Wiley, J. C. On heat loading, novel divertors, and fusion reactors. *Physics of plasmas 14* (2007), 072502.

[75] Kukushkin, A. S., Pacher, H. D., Kotov, V., Pacher, G. W., and Reiter, D. Finalizing the ITER divertor design: The key role of SOLPS modeling. *Fusion Engineering and Design 86* (2011), 2865–2873.

[76] Kukushkin, A. S., Pacher, H. D., Kotov, V., Reiter, D., Coster, D. P., and Pacher, G. W. Effect of the dome on divertor performance in ITER. *Journal of Nuclear Materials 363–365* (2007), 308–313.

[77] Kukushkin, A. S., Pacher, H. D., Loarte, A., Komarov, V., Kotov, V., Merola, M., Pacher, G. W., and Reiter, D. Analysis of performance of the optimized divertor in ITER. *Nuclear Fusion 49* (2009), 075008.

[78] Kuruvila, G., Ta'asan, S., and Salas, M. D. Airfoil design and optimization by the one-shot method. In *33rd Aerospace Sciences Meeting and Exhibit, AIAA paper 95-1478* (1995).

[79] Lions, J. L. *Optimal Control of Systems Governed by Partial Differential Equations.* Springer-Verlag, 1971.

[80] Maddison, G. P., and Reiter, D. Recycling source terms for edge plasma fluid models and impact on convergence behaviour of the braams "b2" code. Tech. rep., Report Forschungszentrum Juelich, JUEL-2872, ISSN 0944-2952, 1994.

[81] Marchand, R., and Dumberry, M. CARRE: a quasi-orthogonal mesh generator for 2D edge plasma modelling. *Computer Physics Communications 96* (1996), 232–246.

[82] Mohammadi, B., and Pironneau, O. Applied optimal shape design. *Journal of Computational and Applied Mathematics 149* (2002), 193–205.

[83] Mohammadi, B., and Pironneau, O. Shape optimization in fluid mechanics. *Annual Review of Fluid Mechanics 36* (2004), 255–279.

[84] Nadarajah, S. K., and Jameson, A. A comparison of the continuous and discrete adjoint approach to automatic aerodynamic optimization. In *AIAA paper 2000-0667* (2000).

[85] Nadarajah, S. K., and Jameson, A. Studies of the continuous and discrete adjoint approaches to viscous automatic aerodynamic shape optimization. In *AIAA paper 2001-2530* (2001).

[86] Neuhauser, J., Bessenrodt-Weberpals, M., Braams, B. J., Carlson, A., Chodura, R., Fussmann, G., Giannone, L., Baas, G., Janeschitz, G., Kaufmann, M., Krech, M., Lackner, K., McCormick, K., Meisel, D., Müller, E. R., Murmann, H., Niedermeyer, H., Poschenrieder, W., Roth, J., Rudyj, A., Schneider, U., Schneider, W., Siller, G., Stabler, A., Steuer, K.-H., Taglauer, E., Tsois, N., Verbeek, H., Vollmer, O., Wunderlich, R., and Würz, H. Tokamak edge modeling and comparison with experiment in ASDEX. *Plasma Physics and Controlled Fusion 31*, 10 (1989), 1551–1568.

[87] Nocedal, J., and Wright, S. J. *Numerical Optimization.* Springer, New York, 2006.

[88] Pacher, G. W., Pacher, H. D., Janeschitz, G., Kukushkin, A. S., Kotov, V., and Reiter, D. Modelling of DEMO core plasma consistent

with SOL/divertor simulations for long-pulse scenarios with impurity seeding. *Nuclear Fusion 47* (2007), 469.

[89] Papadimitriou, D. I., and Giannakoglou, K. C. A continuous adjoint method with objective function derivatives based on boundary integrals, for inviscid and viscous flows. *Computers & Fluids 36* (2007), 325–341.

[90] Patankar, S. V. *Numerical Heat Transfer and Fluid Flow.* Hemisphere Publishing Corporation, Washington, 1980.

[91] Pironneau, O. On optimal shapes for Stokes flow. *Journal of Fluid Mechanics 70*, 2 (1973), 331–340.

[92] Pironneau, O. *Optimal Shape Design for Elliptic Systems.* Springer Series in Computational Physics. Springer-Verlag, New York, 1984.

[93] Pitts, R., Buttery, R., and Pinches, S. Fusion: the way ahead. *Physics World 19* (2006), 20–26.

[94] Rehker, S., and Wobig, H. A kinetic model for the neutral gas between plasma and wall. *Plasma Physics 15* (1973), 1083–1097.

[95] Reiter, D. The EIRENE code user manual, November 2009.

[96] Reiter, D., Baelmans, M., and Börner, P. The EIRENE and B2-EIRENE codes. *Fusion Science and Technology 47* (2005), 172–186.

[97] Rensink, M. E., Lodestro, L., Porter, G. D., Rognlien, T. D., and Coster, D. P. A comparison of neutral gas models for divertor plasmas. *Contributions to Plasma Physics 38*, 1-2 (1998), 325–330.

[98] Reuther, J., and Jameson, A. Aerodynamic shape optimization of wing and wing-body configurations using control theory. In *33rd Aerospace Sciences Meeting and Exhibit, AIAA paper 95-0123* (1995).

[99] Reuther, J., Jameson, A., Farmer, J., Martinelli, L., and Saunders, D. Aerodynamic shape optimization of complex aircraft configurations via an adjoint formulation. In *34th Aerospace Sciences Meeting and Exhibit, AIAA paper 96-0094* (1996).

[100] Robbins, H., and Monro, S. A stochastic approximation method. *The Annals of Mathematical Statistics 22*, 3 (1951), 400–407.

[101] Rognlien, T., Milovich, J., Rensink, M., and Porter, G. A fully implicit, time dependent 2-D fluid code for modeling tokamak edge plasmas. *Journal of Nuclear Materials 196–198*, 0 (1992), 347–351.

Proceedings of the Tenth International Conference on Plasma-Surface Interactions in Controlled Fusion Devices.

[102] Rognlien, T. D., Brown, P. N., Campbell, R. B., Kaiser, T. B., Knoll, D. A., McHugh, P. R., Porter, G. D., Rensink, M. E., and Smith, G. R. 2-D fluid transport simulations of gaseous/radiative divertors. *Contributions to Plasma Physics 34*, 2-3 (1994), 362–367.

[103] Rozhansky, V., Kaveeva, E., Molchanov, P., Veselova, I., Voskoboynikov, S., Coster, D., Counsell, G., Kirk, A., Lisgo, S., the ASDEX-Upgrade Team, and the MAST Team. New b2solps5.2 transport code for h-mode regimes in tokamaks. *Nuclear Fusion 49* (2009), 025007.

[104] Rozhansky, V., Molchanov, P., Veselova, I., Voskoboynikov, S., Kirk, A., Fishpool, G., Börner, P., Reiter, D., and Coster, D. Modeling of the edge plasma of MAST Upgrade with a Super-X divertor including drifts and an edge transport barrier. *Plasma Physics and Controlled Fusion 55* (2013), 035005.

[105] Rozhansky, V., Voskoboynikov, S., Kaveeva, E., Coster, D., and Schneider, R. Simulation of tokamak edge plasma including self-consistent electric fields. *Nuclear Fusion 41* (2001), 387.

[106] Ryutov, D. D. Geometrical properties of a snowflake divertor. *Physics of plasmas 14* (2007), 064502.

[107] Ryutov, D. D., Cohen, R. H., Rognlien, T. D., and Umansky, M. V. The magnetic field structure of a snowflake divertor. *Physics of plasmas 15* (2008), 092501.

[108] Schmidt, S. *Efficient Large Scale Aerodynamic Design Based on Shape Calculus.* PhD thesis, Fachbereich iV der Universität Trier, 2010.

[109] Schmidt, S., Ilic, C., Schulz, V., and Gauger, N. R. Airfoil design for compressible inviscid flow based on shape calculus. *Optimization and Engineering 12* (2011), 349–369.

[110] Schmidt, S., and Schulz, V. Shape derivatives for general objective functions and the incompressible Navier-Stokes equations. *Control and Cybernetics 39* (2010), 677–713.

[111] Schulz, V., and Schillings, C. On the nature and treatment of uncertainties in aerodynamic design. *AIAA Journal 47*, 3 (2009), 646–654.

[112] Schulz, V., and Schillings, C. *Management and Minimisation of Uncertainties and Errors in Numerical Aerodynamics.* Springer-Verlag, Berlin, 2013, ch. Optimal Aerodynamic Design under Uncertainty, pp. 297–338.

[113] Simonini, R., Corrigan, G., Radford, G., Spence, J., and Taroni, A. Models and numerics in the multi-fluid 2-D edge plasma code EDGE2D/U. *Contributions to Plasma Physics 34* (1994), 368–373.

[114] Sokolowski, J., and Zolésio, J.-P. *Introduction to Shape Optimization: Shape Sensitivity Analysis.* Springer'-Verlag, Berlin, 1992.

[115] Spanier, J., and Gelbard, E. M. *Monte Carlo Principles and Neutron Transport Problems.* Dover Publications, Mineola, 1969.

[116] Stangeby, P. C. *The Plasma Boundary of Magnetic Fusion Devices.* Institute of Physics Publishing, Bristol and Philadelphia, 2000.

[117] Stotler, D. P., Karney, C. F. F., Rensink, M. E., and Rognlien, T. D. Coupling of parallelized DEGAS 2 and UEDGE codes. *Contributions to Plasma Physics 40* (2000), 221–226.

[118] Switsers, J. Geometric optimization of fusion reactors. Master's thesis, Departement Mechanica, Faculteit Ingenieurswetenschappen, KU Leuven, 2011.

[119] Ta'asan, S. Pseudo-time methods for constrained optimization problems governed by PDE. Tech. rep., Institute for Computer Applications in Science and Engineering, 1995.

[120] Ta'asan, S., Kuruvila, G., and Salas, M. D. Aerodynamic design and optimization in one shot. In *30th Aerospace Sciences Meeting, AIAA paper 92-0025* (1992).

[121] Tröltzsch, F. *Optimal Control of Partial Differential Equations: Theory, Methods and Applications.* AMS, Providence, 2010.

[122] Tsitrone, E., Loarer, J. B. T., Ghendrih, P., Loarte, A., Andrew, P., Coad, P., Fundamenski, W., Stamp, M., Coster, D., and contributors to the JET EFDA work programme. Particle recirculation studies in JET. *Plasma Physics and Controlled Fusion 44* (2002), 701–715.

[123] Umansky, M., Rognlien, T., Fenstermacher, M., Borchardt, M., Mutzke, A., Riemann, J., Schneider, R., and Owen, L. Modeling of localized neutral particle sources in 3D edge plasmas. *Journal of*

Nuclear Materials 313–316 (2003), 559–563. Plasma-Surface Interactions in Controlled Fusion Devices 15.

[124] Valanju, P. M., Kotschenreuther, M., and Mahajan, S. M. Super X divertors for solving heat and neutron flux problems of fusion devices. *Fusion Engineering and Design 85* (2010), 46–52.

[125] Vercammen, B. Advanced coupling techniques for the plasma edge simulations in fusion reactors. Master's thesis, Department of Mechanical Engineering, KU Leuven, 2010.

[126] Verstraete, T. Multidisciplinary optimization of turbomachinery components using differential evolution. In *VKI Lecture Series: Introduction to Optimization and Multidisciplinary Design in Aeronautics and Turbomachinery* (2010).

[127] Vold, E. L., Prinja, A. K., Najmabadi, F., and Conn, R. W. Coupled plasma-neutral fluid transport computations using the neutral diffusion approximation. *Fusion Technology 22* (1992), 208–226.

[128] Wiesen, S. EDGE2D/EIRENE code interface report. Tech. rep., Forschungszentrum Juelich, 2006.

[129] Wising, F., Knoll, D. A., Krasheninnikov, S. I., Rognlien, T. D., and Sigmar, D. J. Simulation of detachment in ITER-geometry using the UEDGE code and a fluid neutral model. *Contributions to Plasma Physics 36*, 2-3 (1996), 309–313.

Curriculum Vitae

Wouter Dekeyser
°16.02.1986

Education

- 2009–2014 PhD researcher at the Department of Mechanical Engineering, KU Leuven, Belgium.
- 2007–2009 Master in Mechanical Engineering (graduated Summa cum laude), KU Leuven, Belgium.
- 2004–2007 Bachelor in Mechanical-Electrotechnical Engineering (graduated Summa cum laude), KU Leuven, Belgium.

Grants

- PhD Fellow of the Research Foundation Flanders (FWO Vlaanderen)

Experience

- B2 (SOLPS4.3) – B2.5 (SOLPS5.2) code benchmarking in collaboration with Prof. dr. S. Voskoboynikov, St. Petersburg State Polytechnic University, Russia
- Contributions to physics and code speedup for B2 simulations extended to the main chamber wall
- Teaching assistant at Department of Mechanical Engineering, KU Leuven
    - Tutor of several Master theses in edge plasma modeling (2009–2013)

  - Numerical modeling in engineering (2009–2012)
  - Nuclear reactor physics (2009)
- Student researcher at KU Leuven
  - Feasibility study of novel wave energy recuperation system (spring 2009)
  - Trajectory optimization for cam-follower systems (summer 2007)

List of Publications

Articles in internationally reviewed academic journals

1. Dekeyser, W., Reiter, D., Baelmans, M. (2012). Optimal Shape Design for Divertors. International Journal of Computational Science and Engineering, in press.
2. Dekeyser, W., Reiter, D., Baelmans, M. (2012). Divertor Design through Shape Optimization. Contributions to Plasma Physics, 52 (5–6), 544–549.
3. Baelmans, M., Börner, P., Dekeyser, W., Reiter, D. (2011). Tokamak plasma edge modeling including the main chamber wall. Nuclear Fusion, 51, 083023.
4. Dekeyser, W., Baelmans, M., Reiter, D., Börner, P., Vladislav, K. (2011). 2D edge plasma modeling extended up to the main chamber. Journal of Nuclear Materials, 415, S584–S588.

Papers at international scientific conferences and symposia, published in full in proceedings

1. Dekeyser, W., Reiter, D., Baelmans, M. (2012). Adapting computational optimization concepts from aeronautics to nuclear fusion reactor design. EPJ Web of Conferences: Vol. 33. European Energy Conference. Maastricht, 17–20 April 2012 (art.nr. 03009) (pp. 1–8) EDP Sciences.

Meeting abstracts, presented at international scientific conferences and symposia, published or not published in proceedings or journals

1. Baelmans, M., De Schutter, J., Dekeyser, W., Blommaert, M., Reiter, D. (2013). Efficient parameter estimation in 2D transport models based on an

adjoint formalism. 8th Workshop on Fusion Data Processing, Validation and Analysis, Gent, Belgium, 4–6 November 2013.

2. Blommaert, M., Dekeyser, W., Baelmans, M., Gauger, N., Reiter, D. (2013). An automated approach to magnetic divertor configuration design. Aachen Conference on Computational Engineering Science, RWTH Aachen, Germany, 9–11 September 2013.
3. Dekeyser, W., Reiter, D., Baelmans, M. (2012). Shape Sensitivity Analysis of Divertor Target Power Load. 27th Symposium on Fusion Technology. Luik, Belgium, 24–28 September 2012.
4. Dekeyser, W., Reiter, D., Baelmans, M. (2012). Optimal Shape Design for Divertors. 2nd International Conference on Computational Engineering. Darmstadt, Germany, 4–6 October 2011.
5. Dekeyser, W., Reiter, D., Baelmans, M. (2012). Divertor Design through Shape Optimization. 13th International Workshop on Plasma Edge Theory in Fusion Devices. South Lake Tahoe, CA, USA, 19–21 September 2011.
6. Dekeyser, W., Baelmans, M., Reiter, D., Börner, P., Vladislav, K. (2011). 2D edge plasma modeling extended up to the main chamber. 19th International Conference on Plasma-Surface Interactions. San Diego, CA, USA, 24–28 May 2010.

Technical reports

1. Dekeyser, W., Baelmans, M., Voskoboynikov, S., Rozhansky, V., Reiter, D., Wiesen, S., Kotov, V., Börner, P. (2011). B2–B2.5 Code Benchmarking, 52 pp. D-52425 Jülich, Germany: FZ Jülich GmbH, ISSN 0944–2952 (available from: http://juwel.fz-juelich.de:8080/dspace/).
2. Salmagne, C., Reiter, D., Baelmans, M., Dekeyser, W. (2012). Verifikation und Anwendung des B2-EIRENE Codes (SOLPS-ITER) für magnetisierte Plasmen in zylindrischer Anordnung der PSI-2 Anlage, 106 pp. D-52425 Jülich, Germany: FZ Jülich GmbH, ISSN 0944-2952 (available from: http://juwel.fz-juelich.de:8080/dspace/).

Other

1. Dekeyser, W., Manconi, M., Naets, F., Peiffer, V., Reybrouck, J., Van Beveren, K. (2006). De Powerball ontmanteld. Het Ingenieursblad, 75 (10), pp. 41–45.

FACULTY OF ENGINEERING SCIENCE
DEPARTMENT OF MECHANICAL ENGINEERING
DIVISION TME
Celestijnenlaan 300A box 2421
3001 LEUVEN
Wouter.Dekeyser@mech.kuleuven.be